## AN ABSTRACT OF THE THESIS OF

Mark T. Lemke for the degree of Master of Science in Mechanical Engineering presented on January 6, 2017.

Title: Ontologies to Support Repeatable and Valid Customer Requirements Generation

Abstract approved: _______________________________________________________

Robert B. Stone

Due to the growing complexity, sophistication, and interconnectedness of present day engineered products and services, the use of ontologies in engineering design is becoming more widespread. An ontology provides a standardized way of describing concepts in a domain of interest and the relationships between these concepts to better understand the domain as a whole. The benefits of using ontologies vary greatly depending on where and how they are implemented in the engineering space, however the field of complex systems and services can benefit from the level of specification that ontologies provide by reducing the ad hoc nature of some design activities. This paper presents the techniques involved in developing and using two different ontologies both in the aerospace and education domain, each yielding different results. Working with NASA Ames Research Center, this research has improved their customer requirement formulation process. In the education sector, the use of an ontology has helped better understand the parallels between design education disciplines and engineering design curricula. The ontologies explained in this paper were created using unique methodologies, Excel spreadsheets, Python programing, and Protégé Owl, an open source ontology editor. This thesis can be used as a stepping-stone for other engineers who want to achieve a more organized, effective, and productive engineering space.

©Copyright by Mark T. Lemke
January 6, 2017
All Rights Reserved

Ontologies to Support Repeatable and Valid Customer Requirements Generation

by
Mark T. Lemke

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented January 6, 2017
Commencement June 2017

Master of Science thesis of Mark T. Lemke presented on January 6, 2017

APPROVED:

Major Professor, representing Mechanical Engineering

Head of the School of Mechanical, Industrial and Manufacturing Engineering

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Mark T. Lemke, Author

# PUBLICATION THESIS OPTION

This thesis is presented in accordance with the Manuscript Document Format option. Two manuscripts are provided. The first was accepted for publication in the ASME 2016 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference Proceedings. The second paper is prepared for submission to the ASME 2017 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference Proceedings.

ACKNOWLEDGEMENTS

The author expresses sincere appreciation…

- First, I would like to express great gratitude and appreciation towards my awesome parents. Thank you for your continuing support through school and life! I am such a lucky son to have such great parents!
- I thank my beach volleyball and disc golf community! There were many days when I just needed to hit a ball really hard or throw a disc really far so thanks for being there you guys!
- I thank my advisor Rob Stone for guiding me through my graduate schooling. Thanks for always finding time to answer my questions and for letting me wander the world for a year.
- I thank all the guys and gals in the Design Engineering lab for always being cheerful and drinking my homebrew kombucha. One last time, “Go Beavs!”
- I thank Tom Weller for being the best housemate, cook, and bocce player I could ask for. Whenever I am back in town I am coming a knocking.
- Last but not least, I need to send a huge thank you to NASA Ames Research Center for funding this research.

# CONTRIBUTION OF AUTHORS

Anthony A. Nix, Ryan M. Arlitt Ph.D, and Robert B. Stone Ph.D all assisted with the construction of the 2016 DETC paper.

Robert B. Stone also assisted with editing and background content for the 2017 DETC paper.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

# LIST OF FIGURES

## LIST OF TABLES

## LIST OF APPENDIX TABLES

## *GENERAL INTRODUCTION*

The overall objective of this thesis is to leverage the use of ontologies and taxonomies in the customer requirement domain. In philosophy, the word ontology is the study of the kinds of things that exist [1]. However, in the engineering domain ontologies are a classification of theories about the sorts of objects, properties of objects, and relations that exist between these objects in relation to a domain of interest [1]. The backbone of most ontologies typically consists of a hierarchical classification or categorization system, which is known as a taxonomy. Finding and labeling arbitrary complex relations that exist between concepts in a taxonomy and applying these findings to a need yields an ontology.

Early in the design process, a common issue arises during the customer requirement (CR) formulization phase. Because these CRs are usually communicated through natural language, several common issues surface, which include: ambiguity, incompleteness, understandability, testability, and over specificity [2]. These issues need to be resolved in order to save time, money, and accuracy. Because of this, there is a gap in the customer requirement formulation research field that needs further exploration. The research completed in this thesis attempts to alleviate these above mentioned CR formulation problems by using the power of taxonomies and ontologies.

Two application areas for this ontology research were examined, design education and aerospace complex systems, in order to learn more about the capabilities of ontologies. The demands for the application of an ontology or taxonomy were different for both application areas and are listed below.

Design Education

- How are other design disciplines teaching design compared to mechanical engineering?
- What additional techniques could be advantageous to engineering design students to improve their design skills, design process knowledge, and softer skills such as team communication?
- How can current course attributes be located and categorized in the following disciplines: apparel design, industrial design, graphic design, architecture, and mechanical engineering?

Aerospace Complex Systems

- What relationships exist between the software and hardware level of guidance and control CRs?
- How can ambiguity, confusion, and uncertainty be minimized during the CR formulation process?
- In what ways can CRs be categorized and stored in order to improve organization and solution generation?

Some similarities and differences exist between these application areas. Both fields include the notion of involving a customer. For the design education field, the customer would be a mechanical engineering student who is interested in learning about new design techniques that other courses are currently implementing in order to improve their own education or an industry stakeholder who is looking for future employees who have a broader design synthesis capability. In the aerospace complex system sector, the notion of a customer was portrayed in the customer requirement statements that were studied in order to create a better customer requirements formulation process. This notion of customer in this case notably has a more technical focus, but there remains significant room for ambiguity to exist. One difference between these application areas is the source of information used in the research. In the education sector, the authors had to decide on what data would be used to create a taxonomy and how that information would be searched and collected. However, for the aerospace domain, the author was presented with 20 guidance and control CR's by NASA Ames Research Center to study. Because of these differences, various ontology generation approaches needed to be performed and they present the ability to validate the approaches for markedly different audiences.

A conference paper was written for each of these application areas in order to report the findings of each. Both papers are included in this thesis with detailed description of the methodologies followed to create a taxonomy and ontology that addresses the above demands. Ontology and taxonomy creation methods were researched extensively and are outlined in each background section. It turns out that ontologies can be very powerful!

*FIRST MANUSCRIPT*

# DESIGN EDUCATION ACROSS DISCIPLINES: OPPORTUNITIES FOR CURRICULUM ADVANCEMENT

Mark T. Lemke, Anthony A. Nix, Ryan M. Arlitt Ph.D, Robert B. Stone Ph.D

Proceedings of the ASME 2016 International Design Engineering Technical Conferences
& Computers and Information in Engineering Conference
IDETC/CIE 2016
Aug 21-24, 2016, Charlotte, North Carolina, USA

**DETC2016 - 59767**

## ABSTRACT

Design education is a large field. It is not just limited to engineering design but can also include apparel design, industrial design, graphic design, architecture, and many others. These disciplines instruct students to follow a similar design process to what is generally taught in engineering design. However, these other disciplines contain a variety of instructional techniques, class structures, and class types that are not regularly included in engineering design. While design engineers tend to get a background rich in math and science, instructing students in design can be difficult. Many of these math and science classes focus on one approach and one right answer. However, in design the answers tend to fall on a spectrum from unsatisfactory to varying levels of satisfactory to ideal and innovative solutions, all of which can be uncovered using widely varying design methods. Despite the rigidness of the mechanical engineering curriculum there are areas where the implementation of techniques used in the other design disciplines could be advantageous to help engineering design students improve students design skills, design process knowledge, and softer skills such as team communication. The research done in this paper examines how the curricula of design disciplines could influence the coursework of students focusing on the design area of mechanical engineering.

## INTRODUCTION

Recently, researchers have noted a decrease in the number of available design credits [3], an emergence of topical specialty areas in engineering (e.g., mechatronics) competing for credit hours [3], and a lack of engineering coursework emphasizing creative aspects of higher-level learning [4, 5]. This is concerning since design courses are where students use their knowledge of mechanics, control systems, and other traditional engineering coursework to create new products, processes, and systems. A three year longitudinal study focusing on how engineering students approached the design process found differences based on the amount of design coursework the students had previously completed [6-9]. The instruction of and knowledge gained by applying these learned principles of physics should be a valued portion of the engineering curriculum.

Design education is larger than just engineering design. It includes disciplines such as industrial design, graphic design, architecture, apparel design, interaction design, and others. These other design fields are sometimes considered more “creative” than engineering design, yet they often include the instruction of a design process that is currently or was at one time similar to the typical engineering design process put forth in engineering design texts [10-13]. Despite teaching a similar process, the method of instruction and instructional setting in which the process is taught varies among disciplines.

## Motivation

The motivation for this study stems from the authors’ desire to understand what other design disciplines are teaching and how it is being taught in order to provide a blueprint for an improved engineering design education. Locally, the Oregon State University (OSU) apparel design program teaches a design process that is very similar to engineering design, yet the apparel design curriculum is very different than the mechanical engineering design curriculum. Not only are the course topics different (as expected), but the manner in which those topics are instructed is also different. This begs the question: can engineering design instruction benefit from the design instruction techniques in other disciplines? This research begins to examine this question by examining the differences between the mechanical engineering course content and instructional methods as compared to other design disciplines.

# BACKGROUND

## Design Education

Design education has been a topic of discussion since early in the 20th century [14] and continues to this day. While viewpoints on how design should be conducted have changed over time, the need to effectively communicate design concepts and to instruct students on how to approach design still persist. Some key elements to creative engineering classes have been applied by comparing and contrasting engineering education in multiple cultures. These elements include: healthy social activities, clear goals, appropriate problems, continuous improvement, and the engagement of the teaching staff [15]. Other research has found effective learning includes powerful experiences that are not typically available in traditional classroom settings such as the Warman Design and Build competition [18], or experiential project based lab sections which increase students's enjoyment and engagement [19].

One current approach to introducing students to the methods used in other design disciplines has been the creation of interdisciplinary graduate programs and multi discipline product design programs [20]. These programs exist to expand student involvement with other disciplines such as industrial design, architecture, law, and others [21]. In some programs this interdisciplinary inclusion has been used across courses, across terms, and across disciplines to provide experience in interdisciplinary team interaction. This process has been met with improved student enjoyment [20].

While disciplines such as apparel design have attempted to incorporate engineering design techniques [21, 22], there has been little movement with engineering design attempting to incorporate the instructional techniques of the other design disciplines. The main area of crossover centers around applying a studio classroom model often associated with disciplines such as architecture and industrial design, to engineering design [23, 24]. This instructional style has yet to be adopted by the majority of universities teaching engineering design, however.

It should be noted that other instructional techniques from other disciplines are not as easy to translate to engineering. For example, students without the proper background and understanding of physics may use design as a way to express their own creativity instead of using design to resolve a set of complex restrictions and issues [25]. Another limitation with implementing many design courses is the ability of faculty, students, and the educational facility to support this instructional change efficiently. Research has been conducted on determining an effective method to introduce this style of instruction and learning to the engineering curriculum [26]. The authors found that ideally a student's time in an engineering program would contain design projects in four quadrants they proposed: team-content, team-process, individual-content, and individual-process. This amounts to a design oriented class every year in a 4-year curriculum [26].

## Taxonomies for Design Education

When examining the different, yet related, fields of design education, a common language and thoughtful classification scheme is needed. Generally, a taxonomy or ontology supports this need. A taxonomy is a classification and naming of entities in an ordered system that is intended to indicate natural relationships. As background, this section will explore different approaches and classifications researchers have taken in order to create taxonomies for their research.

There is an ongoing discussion in the taxonomy community whether taxonomies should be created bottom up, top down, or from the middle out. Sowa used the bottom up approach, which starts with a list of the most specific, detailed entities then groups these words into appropriate categories [27]. Vice versa, a top down approach would start by identifying the most general concepts and working downward to more specific concepts [28]. Uschold and Gruninger make an argument with their taxonomy that the middle layer of concepts should serve as a starting point and the development proceed in both directions [29].

Using an inspirational or motivating approach to taxonomy design starts with the premise about why a taxonomy is needed. The developer uses their individual imagination, creativity, and personal views on the domain of interest in order to meet the recognized need. Grüninger found motivation from existing problems that were not adequately expressed in existing taxonomies in order to create their taxonomy, TOVE [29]. These researchers believe that any new taxonomy must first describe a motivating, personal scenario and set of intended solutions to the problem being analyzed. Similar to the inspirational approach, a collaborative approach is a joint effort reflecting experiences and viewpoints of several individuals who work together. Chances of a very descriptive, detailed taxonomy being produced are high due to the diversity and depth of knowledge of the team as a whole.

Alternatively, inductive and deductive approaches have been used to create taxonomies. With an inductive approach, developers observe, examine, and analyze specific cases in the domain of interest. The resulting taxonomy characterizations of one specific case can then be applied to similar cases in the same domain [30]. Many times these taxonomies can then be connected which can yield a more complicated system called an ontology. A deductive approach to taxonomy design applies adopted general principles towards a specific case in examination. This method involves assigning predetermined characteristics or labels to a list of domain entities being studied [32].

Dahlgren's ontology and the Plinius project both use a text corpus approach as the basis for their development [31, 32]. This approach uses computer programs as a starting point to search a collection of written texts or databases in order to detect common themes, relevant concepts, or popular words. The corpus approach can yield exact entities that can be used in a taxonomy or create inspiration to the developer.

Taxonomy organizations can be broken down into three major approaches. First, the most common taxonomy organization is a single tree-like concept hierarchy with multiple inheritances. Usually a "is-a" relationship is used to link the hierarchy tree together. Opposite to this, a distinction approach entails parallel dimensions along which one or more top-level categories are subcategorized [28]. This can be visualized as many tree-like hierarchy next to each other with non-related top-level categories. The third approach to taxonomy organization is including many small local taxonomies that are linked together by relations or axioms [28]. This approach is usually used if an ontology is sought after as the end result.

After considering the scope and motivation of this research project, the authors believe that a combination of top-down, inspirational, deductive, corpus, and distinction approaches should be incorporated in the creation of this class course taxonomy.

# RESEARCH

## Research Questions

The main research question driving this paper was *How are other design disciplines teaching design compared to mechanical engineering?* Once this question is answered one can apply this knowledge to the mechanical engineering curriculum and attempt to make improvements based on these results.

## Data Collection

The first step in data collection was deciding on universities to examine. The process for determining which schools to include was based on expert opinion and various internet published rankings of the disciplines being examined. These rankings were provided by organizations such as U.S.News & World Report, Industrial Designers Society of America, startclass, arch daily, and others. Expert opinion was most important for the mechanical engineering group. Each school in this group required a strong design program, and not many rankings for mechanical engineering programs include a ranking by design specialty – they rank the program as a whole including robotics, thermodynamics, and other areas.

The selection is not meant to represent the top five universities in each discipline but instead to represent five well respected universities with solid programs in the disciplines being examined. The selection also excludes small specialty design schools such as Rhode Island School of Design due to the general lack of course breadth offered by small universities. Carnegie Mellon is the smallest university sampled at 13,285 students. The complete list of universities sampled and their populations are included in Table 1.

Table 1: List of Universities Surveyed

| Discipline | School | | | | |
|---|---|---|---|---|---|
| Mechanical Engineering | Oregon State University | Stanford University | Carnegie Mellon University | University of Texas-Austin | University of Michigan |
| Apparel Design | Oregon State University | University of Minnesota | Cornell University | Kent State University | Iowa State University |
| Industrial Design | Carnegie Mellon University | Ohio State University | University of Illinois | University of Michigan | Arizona State University |
| Architecture | Virginia Polytechnic Institute and State University | Iowa State University | Georgia Institute of Technology | University of Michigan | University of Texas-Austin |
| Graphic Design | Carnegie Mellon University | University of Michigan | University of Illinois | Virginia Polytechnic Institute and State University | Pennsylvania State University |

The next step was to identify the course list for each program sampled at each university. Courses examined included all courses (including the design courses) being offered from the full course catalog of each discipline via the universities' online course catalogs. For example, mechanical engineering classes such as numerical methods, control systems, and instrumentation were included in the analysis. As a general rule, classes outside of the discipline were not considered to provide a boundary.

The research only includes courses that are available to undergraduate students. This includes graduate level classes that are open to undergraduate students. Those that are specialized beyond the undergraduate level were not included because not all programs contain a graduate program in the discipline examined. This limitation also keeps the course list to a manageable level for analysis.

The class lists were loaded into a spreadsheet with the credit hours or units that each class was worth. This spreadsheet was then used so the authors coding the classes had an identical base.

## Taxonomy Generation

In order to compare the various classes across disciplines, a taxonomy of course types was needed. For this work we use a mixed data-driven and theory-driven approach to taxonomy creation.

For the initial data-driven approach, candidate taxonomy terms were collected using word frequency analysis. This process used the Natural Language Toolkit (NLTK) [33] in the Python programming language in order to perform to basic text preprocessing and analysis. The course descriptions of two universities from each of the five program types (10 schools total) were found online and copied into one text file. Once this file was imported into Python, various NLTK functions were used to remove stop-words (high-frequency words that usually have little lexical content), three letter words, and punctuation. After capitalizing all of the remaining words, a word frequency distribution was performed which yielded a spreadsheet with an ordered list of most frequently used words. The top results for this distribution list can be seen in Table 2.

Due to the inherent challenges and ambiguities associated with solving problems like synonymy (a state of two items of being synonymous) and polysemy (a coexistence of multiple meanings of a word), we do not account for differences in word sense. For example, this approach does not differentiate for the word “material” being used in the senses of “course material” and “structural materials.” These issues were mitigated in the theory-driven part of the taxonomy development by considering the possibility of misleading word frequencies when choosing taxonomy terms.

In the theory-driven stage, an iterative approach was taken to develop the taxonomy. The authors first examined the list of the most commonly used words and grouped them according to topic similarity. These groups were then analyzed for a common broad theme. The first iteration of the taxonomy was then applied to ten classes in each discipline, after which the results and ability of the taxonomy to accurately describe and differentiate the classes in meaningful ways was examined by the authors. Successive iterations were then assembled and evaluated using a similar procedure. The final taxonomy used to evaluate the classes can be found in Table 3.

Table 2: Most Common Words in Course Description

| Most popular words in course description | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Design | 648 | Work | 139 | Problem | 93 | Method | 86 | Develop | 78 |
| Course | 303 | System | 136 | Analysis | 93 | Building | 84 | Class | 73 |
| Student | 255 | Product | 131 | Material | 92 | Form | 83 | Experience | 71 |
| Project | 173 | Process | 116 | Study | 87 | Research | 82 | Visual | 69 |
| This | 143 | Students | 98 | Development | 87 | Engineering | 78 | Architecture | 68 |

Table 3: Taxonomy Created to Code Courses

| Class Type | | | Setting | | |
|---|---|---|---|---|---|
| 1 | Theory | Students learn and apply theory to practice scenarios | 1 | Lab | Students have a hands on focus on testing and learning |
| 2 | Applied Theory | Students apply theories to design problems, often involves a project | 2 | Studio | Hands on discussion with aspects of theory, designing, and prototyping |
| 3 | Management | Business oriented, project management | 3 | Lecture | Traditional class held in a large room with minimal student involvement |
| 4 | Design and Build | Project based design and build class with no theory | 13 | Lecture/Lab | Class contains a typical lecture component as well as a separate lab component |
| 5 | Culture | Includes history of design or profession as well as examining the influence of other cultures | | | |
| 6 | Tech Skills | Learning software or how to use tools | | | |
| 7 | Professional Development | Learning about the profession and what people in industry do | | | |

# RESULTS

## Class Assignment

The class assignment phase of the research was completed by two of the authors. The coders were both graduate students in mechanical engineering, one pursuing a M.S. and the other a Ph.D. These two authors evaluated all of the courses examined using solely the online descriptions provided by the schools. Both assigned a class type and setting value to each class that appeared in the class list. This coding was done based off the course descriptions found online. Overall 1518 courses were analyzed. Not all classes were coded by the authors. Special topics classes often did not have a course description and were not classified as they change frequently. Classes such as independent study, reading and conference, and internship were also not included because there is not a way to analyze what is being taught or learned in those settings.

Two coders coded each question independently based off of the course descriptions provided by the university. Cohen's Kappa values for inter rater agreement were used to examine the inter rater agreement of the coding process [34]. These values were calculated for the class type, setting, and the full code. The values calculated were 0.377, 0.407, and 0.367 respectively. A commonly used guideline of 0.21 to 0.40 corresponds to fair agreement [35]. After all of the coding took place the coders met and reconciled any codes they disagreed upon. Therefore the final list of codes used for the analysis was agreed upon by both coders. Section 5.3 of this paper discusses the possible explanations for the values.

## Analysis

The analysis of the data was conducted in two parts: the analysis of the type of class and an analysis of the setting the classes were held in. The comparisons are based on the percentage of credit hours or units (out of the program total) that are assigned by the home institution to a particular type or setting of a course. The percentages were determined by first finding the percentage of type and setting for each school listed under each discipline. After that the averages for each university were combined to generate one average for the discipline. There were no large discrepancies between the universities within a discipline as the standard deviations between the most prominent types of courses ranged from 4% to 15%.

When comparing the engineering curricula to the other design disciplines one will notice differences between them. Engineering curricula have more theory-based coursework than the other disciplines and have less coursework based in the other six types. The second highest percentage of curricula for mechanical engineering lies in the applied theory, however mechanical engineering was behind architecture, industrial design, and graphic design in the percentage of applied theory classes that make up the curricula. Theory and applied theory make up 88% of the total engineering curricula, leaving very little opportunity for the other types of coursework.

The most often used setting for teaching engineering takes place in the lecture. The lecture is followed closely to a lecture/laboratory setting which includes separate lecture and laboratory units instead of a studio type class which typically combines those components. These results can be found in Table 4 as well as Figures 1 and 2.

Table 4: Percentage Make Up of Design Curricula

| | Mech. Eng | Apparel Design | Industrial Design | Architecture | Graphic Design |
|---|---|---|---|---|---|
| Theory | 70% | 45% | 36% | 45% | 33% |
| Applied Theory | 18% | 23% | 31% | 28% | 28% |
| Management | 2% | 2% | 1% | 0% | 0% |
| Design & Build | 1% | 3% | 5% | 0% | 7% |
| Culture | 0% | 13% | 5% | 14% | 4% |
| Tech Skills | 7% | 11% | 17% | 6% | 23% |
| Professional Development | 2% | 2% | 5% | 6% | 5% |

| | Mech. Eng | Apparel Design | Industrial Design | Architecture | Graphic Design |
|---|---|---|---|---|---|
| Lab | 2% | 1% | 2% | 8% | 2% |
| Studio | 0% | 33% | 50% | 34% | 55% |
| Lecture | 77% | 61% | 47% | 46% | 43% |
| Lecture/Lab | 20% | 5% | 1% | 12% | 0% |

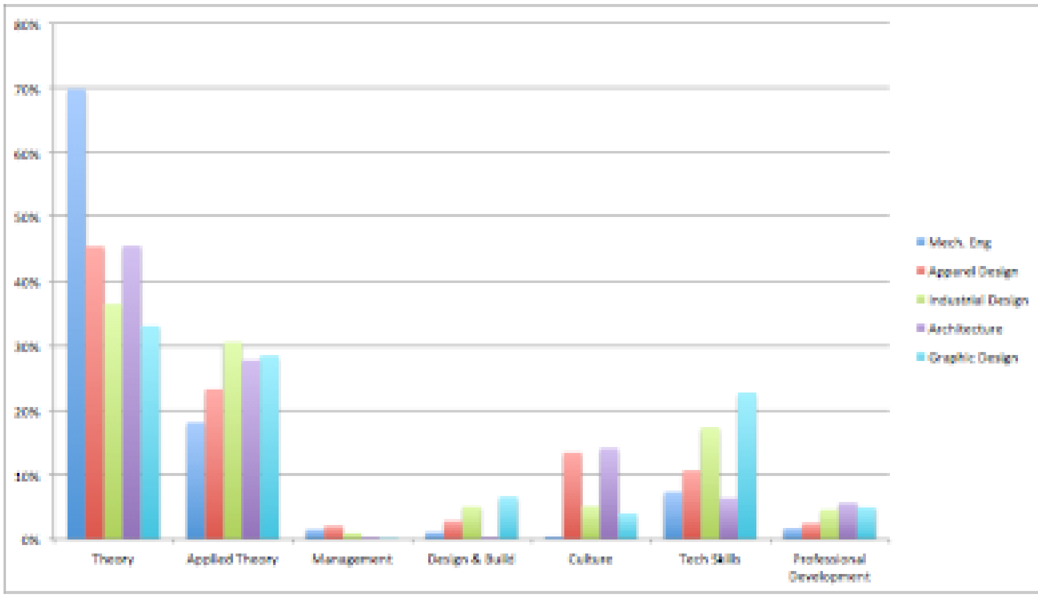


Figure 1: Type Percentage of Coursework Offered
by Design Disciplines

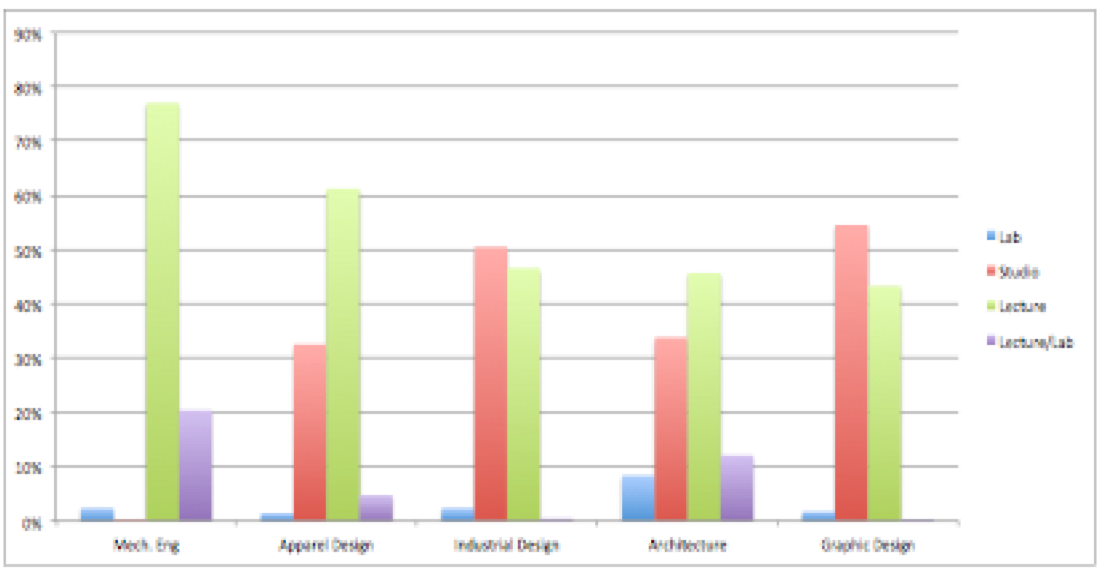


Figure 2: Setting Percentage of Coursework Offered
by Design Disciplines

# DISCUSSION

## Quantitative

One of the main takeaways from the analysis of the data was that engineering coursework is typically a theory based class in lecture format with students demonstrating their knowledge of the topic through working textbook type problems. While this will not come as a surprise to most readers, it does show room for improvement. One area where the number of theory classes could be reduced in a manner beneficial to the students would be in the adoption of more applied theory coursework. In these types of classes students would do less textbook problem solving and more application of learned knowledge to real life projects and problems. This in turn leads to the second takeaway. Over 70% of the engineering curricula examined is taught in a typical lecture based setting with the next largest portion being lecture based with a laboratory component. A currently non-existent studio based component could be effective in turning the heavy book problem based theory courses into an applied theory course with the students actively engaged in their learning.

There was a lack of design-build coursework in mechanical engineering found by the authors. While this may appear at odds with the reader's expectation, there is good reason. This finding is due to the majority of mechanical engineering courses that require a prototyping project involve the instruction and application of design theories and processes and therefore were coded as an applied theory course. There are very few courses identified in the mechanical engineering curriculum where students are provided a design prompt and asked to design and build a prototype without the instruction on the design process they should follow. While some capstone design courses follow this process, others do contain a theory aspect due to a lack of previous design coursework.

Another takeaway relates to the amount of culture based coursework that current mechanical engineers are offered – standing at 0%. While not all other disciplines offer large amounts of culture or history in their discipline-based coursework, they have noticed a need for it and offer multiple courses in the history of their discipline, culture in certain eras, or the culture in certain locations. One question that this finding poses is:

"Should engineering students graduate with some knowledge of the history of engineering or how other cultures practice and view engineering?"

## Application & Advancement

There exist many possibilities on how to apply the findings of this research to the current engineering curriculum both in the adoption of styles into current classes as well as the creation of new classes. The implementation of more studio type classes could be completed harmoniously within the current system. The use of cooperative learning techniques or SCALE UP classrooms [36] could be used to foster non-traditional learning environments. For example, introducing a "hands-on" day during classes such as dynamics where one day a week the class is not based on the book but based on observations and experiments conducted during class. These experiments could include launching projectiles and measuring time and distance traveled to determine velocity. Stress and strain can be measured by bending boards and then asking students to measure deflection given a force, and determining material properties. This is a measured step toward studio model requiring minimal course redesign and gaining the benefit of the interactive learning element. There are multiple ways to implement a history or culture type class within engineering. One implementation could be a class based on the contributions to engineering design from Renaissance figures such as Leonardo da Vinci or a class based on previous engineering failures, the reasons for the failures, and the societal issues caused by those failures. The history of space or automobile engineering could be another course touching on engineering history.

Design and build classes on a five week timeline with a competition is a possible way to implement more design and build classes. Ideally in the first week the students would receive a design prompt for a competition in five weeks. Two of these instances could be completed in a ten week term (quarter) or three in a sixteen week semester. Students may be required to provide documents showing they followed the design process however the instructor's introduction of new material to the course would be minimal as this course is about the students ability to apply what they have learned.

These recommendations are not meant to usurp the current curriculum but to provide options for students to advance their design knowledge and abilities.

## Limitations

One of the limitations of the study is the fair inter rater agreement using Cohen's Kappa. This was corrected by the coders reconciling the codes before completing the analysis. There are multiple issues that contribute to a rating of fair instead of moderate to substantial. One reason for this is the coding was done purely on the course descriptions posted by the university. All courses were analyzed by using the course description from the website. The raters were asked not to project their own opinions about what a course with that title should contain but rather base the coding solely on what was read in the description. Therefore the accuracy of the coding is purely based in the accuracy of the course descriptions supplied. Not all course descriptions were clear as to the setting of the course or the main instructional method being used. This ambiguity could be one of the reasons for the discrepancies during coding.

Not all courses offered by the universities were analyzed. One important set of courses that were not analyzed is those listed under special topics. These courses are often in flux and the special topics moniker is also used for courses in development. Often novel and interesting courses are contained within this category however they were not analyzed due to the amount of turnover special topics classes undergo and not all universities posting what courses were contained within the special topics designation.

One last limitation of the research is due to the way in which universities designate credit hours. The courses were analyzed according to how much of the program's total curriculum they represented as a percentage because of this difference. This was due to differing program requirements and course credit coding schemes at the selected universities. For example the course Dynamics is 3 credits at Oregon State University, 4 units at Stanford University, 10 units at Carnegie Mellon, 3 semester hours at UT-Austin, and 4 hours at the University of Michigan. While this issue was dealt with it adds another layer of complication to the analysis.

## CONCLUSIONS/FUTURE WORK

Engineering design takes a different approach to teaching design than the other design disciplines. This is to be expected because of all the design disciplines engineering is the most based in science. Another reason is the breadth of topics that engineering design covers everything from heat transfer to sustainable product design. However examining and learning from how other design disciplines approach the instruction of their students could prove fruitful. An implementation of more hands on involvement in a studio setting could be beneficial to the students learning and their ability to apply the course topics outside of textbook style problems. Studio classrooms, cooperative learning environments, and SCALE UP classrooms are settings that take advantage of the amount of available technology, tutorials, and activities for students. The freedom students have in a studio type class also helps them think creatively. The structured one right answer style class that is prominent in many engineering courses does not allow for students to work on thinking creatively or allow the opportunity for multiple perspectives.

Another aspect that could be beneficial to students is more time spent practicing or applying engineering principles in a design and build activity.  The engineering curriculum is filled with learning opportunities and classes devoted to increasing the knowledge base of the students passing through the program, in contrast a small amount of time is spent putting these skills together in a design project. These types of courses that increase student engagement and enjoyment have been shown to attract more students to engineering and to attract students from a more diverse population [37, 38].

Their are tradeoffs to implementing more studio type classes and perhaps the most critical one is resources required – in both instructional faculty and physical classroom space. Class sizes may need to be altered and a room where these activities could occur may have to be created. Creating smaller class sizes would increase the number of sections and require more instructors. However these tradeoffs may be mitigated when considering the impact to the student: gains in enjoyment, engagement, diversity, and possible increased creative thinking ability for real world design problems.

Future work on this topic includes implementing and examining more studio and hands on style courses within the mechanical engineering curriculum in an attempt to gain understanding on what the most efficient aspects of a studio type class are. These aspects could then be implemented in other courses. Another route for future work includes the refinement of the taxonomy generated for this study. While proving useful for this work, the taxonomy could be improved upon and eventually expanded to include all courses offered in the university setting. One last area for future work involves the expansion of courses analyzed from those that exist within the discipline to any course that a student in that discipline is likely to enroll in. This would require extensive time and energy devoted to completely understanding any elective and all baccalaureate core classes.

Other design disciplines are doing things differently. As an engineering design community, the authors encourage all design educators to take a step back and review curriculum for ways that techniques in other design disciplines can improve the learning experience for students and the overall product (i.e., engineers ready to tackle real problems) we produce.

*SECOND MANUSCRIPT*

# ONTOLOGIES TO SUPPORT CUSTOMER REQUIREMENT FORMULATION IN AEROSPACE DESIGN

Mark T. Lemke, Robert B. Stone Ph.D

Prepared for Proceedings of the ASME 2017 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2017
Aug 6-9, 2017, Cleveland, Ohio, USA

**DETC2016 – #####**

## ABSTRACT

The major goal of customer requirement formulation is to achieve a common understanding between the project stakeholders and the engineering requirements. Many times, this process can be ambiguous, incomplete, and time consuming especially when more than one engineering discipline is involved. Therefore, adequate requirement formulation tools can be a major contributor to solving these challenges. The use of ontologies provides a standardized way of describing concepts in a domain of interest and the relationships between these concepts to better understand the domain as a whole. This paper describes the methodology used to create an ontology derived from twenty customer requirements of a mid-size, twin-engine, commercial transport-class aircraft provided by NASA Ames Research Center. One key stipulation that NASA had was that this ontology effectively captures the relationships that exist between the hardware and software level of each customer requirement. The final ontology was created using Protégé Owl, an open source ontology editor, which will be used by NASA in order to improve the customer requirement creation phase of future NASA products. This research paper fills a gap in the customer requirement research field by introducing the use of ontologies to reduce ambiguity and repetition.

## INTRODUCTION

This conference paper was motivated by a case study performed by NASA Ames Research Center. NASA is interested in the research of verification technologies, so a case study on a TCM (Transport Class Model) was performed [39]. This flight-critical, unclassified model originated from NASA Langley's Transport Class Model [40], which is a simulator of a mid-size, twin engine, commercial transport-class aircraft, similar to Boeing's 737 [39]. Because the TCM was not intended for embedded production-level code, written requirements needed to be solicited using several resources, which included: pilot training manuals, Federal Aviation Regulations, and subject matter experts [39]. A significant amount of work was involved formalizing and disambiguating these requirements in order for them to be used at the subsystem level to drive the verification approach of the overall system successfully [39]. NASA researchers noticed that CRs are often written in natural language and needed to be translated into a clear notation with unambiguous semantics [39]. Also, the relationships that exist between the software and hardware level needed to be better understood. Due to difficulties during their case study, NASA sponsored a Center for e-Design project (a NSF Industry/Univeristy Cooperative Research Center) through the Design Engineering Lab at Oregon State University to explore formal methods that support unambiguous CR generation. The overall goal of this research project was to enable any two design engineers to understand and write the same customer requirement.

# BACKGROUND

## Ontologies

The word "Ontology" has many different meanings depending on the context and community it is designed for. For the purpose of this paper, an ontology is a classification and naming of entities in an ordered system intended to indicate natural relationships [41]. In other words, an ontology is the study of how one categorizes things and what are the guidelines for each category in order to help organize and efficiently access useful information. In 1993, one of the leading researchers in the ontology field, Gruber, defined the notion of an ontology as an explicit specification of a conceptualization, which every knowledge base, knowledge based system, or knowledge level agent is committed to either explicitly or implicitly [42]. This conceptualization would capture the objects, concepts, and other entities that are assumed to exist in the domain of interest and the interactions between them [42]. McGuinness and Noy argue that one positive result of an ontology is to create a common vocabulary for researchers to use in order to share an understanding of the structure of information among people or technology [43]. Ontologies have been applied to a wide range of domains from cookbooks and pet selection to wine-meal pairing and organic chemistry.

Since the use of ontologies is a rapidly growing field, technology to support the creation and distribution of ontologies is becoming more popular. In the forefront, Protégé, a free open source ontology editor, has been leading the industry for more than a decade [44]. This knowledge management system provides a graphical user interface to create, edit, and share ontologies. The OWL plugin is a Semantic Web extension of the Protégé platform that supports Web Ontology Language [45]. This language is designed to represent rich and complex information about things, groups of things, and the relationships between these things [46]. All of the ontologies created in this paper used Protégé OWL V4.

## Ontology Creation

In order to better understand how current methodologies are developed, organized, and created in this emerging field of science, a literature review was conducted analyzing various approaches and classifications that others have used in developing successful ontologies. There is an ongoing discussion in the ontology community whether ontologies should be created bottom up, top down, or from the middle out. Sowa refined a bottom up approach, which starts with a list of the most specific, detailed entities then groups these words into appropriate categories [27]. Vice versa, a top down approach would start by identifying the most general concepts and working downward to more specific concepts [28]. From a third perspective, Uschold and Gruninger make a compelling argument with their ontology that the middle layer of concepts should serve as a starting point and that development proceeds in both directions [29].

Using an inspirational or motivating approach to ontology design starts with the need of the ontology at the forefront. A developer using this approach puts their individual imagination, creativity, and personal views on the domain of interest in order to meet the recognized need for the ontology. Grüninger found motivation from existing problems present in their field that were not adequately expressed in existing ontologies, in order to create their ontology TOVE [29]. Some researchers believe that any new taxonomy must first describe a motivating and sometimes personal problem and a set of intended solutions to this problem. For this need-based method, the creators must be subject matter experts in the domain of interest. Similar to the inspirational approach, a collaborative approach is a joint effort reflecting experiences and viewpoints of several individuals who work together. Ontologies that are generated using these two approaches often are very detailed and complex due to the diversity and depth of knowledge of the individual or team as a whole.

Alternatively, inductive and deductive approaches have been used to create less complex ontologies. With an inductive approach, developers start by observing, examining, and analyzing specific cases in the domain of interest in order to uncover the principal theories present. After examining the resulting ontology, characterizations of one specific case can then be applied to similar cases in the same domain [30]. Many times these ontologies can then be connected to generate a more complicated system. On the other hand, a deductive approach to ontology design applies adopted general principles or theories towards a specific case under examination. This method involves assigning predetermined characteristics or labels to a list of domain entities being studied [32].

Dahlgren's ontology and the Plinius Project both used a text corpus approach as the basis for their ontology development [31, 32]. This approach uses computer programs as a starting point to search a collection of written texts or databases in order to detect common themes, relevant concepts, or popular words. The corpus approach can yield exact entities that can be used in an ontology or create inspiration in the developer.

Ontology organization can be broken down into three major approaches: single tree-like, distinct, or local. First, the most common taxonomy organization is a single tree-like concept hierarchy with multiple inheritances. Usually a "is-a" relationship is used to link the hierarchy tree together. Opposite of this, a distinction approach entails parallel dimensions along which one or more top-level categories are subcategorized [28]. This can be visualized as many tree-like hierarchies next to each other with non-related top-level categories. The third, local approach to ontology organization includes many small local ontologies that are linked together by relations or axioms [28].

After considering the scope and motivation of this research project, the authors believe that a combination of top-down, inspirational, deductive, and local approaches should be incorporated in the creation of their ontology.

## Customer Requirement Formulation

In engineering design, the needs of the stakeholders are expressed through statements called customer requirements. Often, these requirements are communicated with natural language due to ease of writing and reading. However, several common issues associated with these requirements include, but are not limited to ambiguity, incompleteness, understandability, testability, and over specificity [2]. Because of this, correct customer requirement formulation practices should be followed in order to create clear, concise, and verifiable statements.

Hook states that a well written requirement is necessary, verifiable, and attainable [47]. The major mistake seen in today's requirement writing is the inclusion of non-verifiable subjective words like: easy, hard, heavy, long, loud etc. [47]. The following is a list of the other common problems in writing requirements: making bad assumptions; writing implementation (how) instead of requirements (what); describing operations instead of writing requirements; using incorrect terms, using incorrect sentence structure or bad grammar; missing requirements; and over-simplifying [47]. Robertson and Robertson [48] have developed criteria and resources for all invested parties to follow in order to avoid these common problems. A rationale criterion adds background reason to each requirement while a fit criterion quantifies or measures the requirement [48]. A requirements specification template can be used as an outline and guide to writing requirement specifications [48]. Another resource called a shell or snow card creates opportunity to ensure each requirement has the correct components for that type of requirement [48]. By following these guidelines put forth by Robertson and Robertson while avoiding the most common problems addressed by Hook, an engineer will reduce the chances of confusion.

## RESEARCH METHOD

Below is the step-by-step methodology that was created and followed in order to create the final Customer Requirement Formulation Ontology and Final Customer Requirement Templates. A spreadsheet, text editor, and Protégé Owl were the only programs used during this research. In Protégé Owl, an is-a relationship was followed between all classes and subclasses. A list of acronyms used in this research can be found in Table 5 below. All other tables mentioned in this section are located in Appendix A: Customer Requirement and Entity Tables.

Table 5: Acronyms List

| Acronym | Definition |
|---|---|
| CR(s) | Customer Requirement(s) |
| MCP | Mode Control Panel |
| HC | Heading Control |
| AC | Altitude Control |
| FCC | Flight Control Computer |
| FPA | Flight Path Angle |

## Step 1: Customer Requirement Processing

The first step worked directly with the 20 guidance CRs NASA Ames Research Center provided, which can be seen in Table 6. For this step, each CR was examined and rewritten following correct CR writing practices as outlined by Hook [47] and the Robertsons [48]. Since the authors are not experts with the details of the TCM domain, all edits that were addressed related to either sentence structure or grammar, not content or meaning. These changes included: breaking up one compound CR into two separate CRs, eliminating redundant CRs and words, punctuation and tense agreement, and uniform sentence structure. The new list of CRs that were used for this research can be seen in Table 7, which consists of 26 separate CRs. Of the 20 original CRs, there were seven CRs that remained unchanged, which are denoted by an * in Table 7. The reduction of ambiguity was already visible with the conclusion of this step. Also, because of this pre-processing, the following steps were made easier and less complicated to perform.

## Step 2: System and Relationship Ontology Creation

After the Customer Requirement Processing phase was completed, the next step was to start determining specific entities that would be used in the ontology itself. In order to uncover the existing relationships between the software and hardware level imbedded in each CR, two separate ontologies were created. First, the System Ontology, seen in Figure 3, was created in order to capture all of the hardware, software, and human involvement in each CR. This was done by first creating Table 8, which captured every system that was required or used for each CR. Next, the Relationship Ontology, seen in

Figure 4, was formed by assigning any required communication or action entities between the assigned system entities for each CR. Software to Hardware, Human to Software, Human to Hardware, Software to Software, and Software to Human relationships all existed within the list of 26 CRs and can be examined in Table 9. Both of these ontologies include entities that have a colon included; this denotes a variable that needs to be read by either a human or software system. Again, both authors used their best judgment while assigning entities to their respective ontology.

## Step 3: Goal Ontology Creation

Since every correctly written CR entails a goal or objective, the next step was to capture this information for each of the 26 CRs. Table 10 was then created by inspecting each CR exclusively and deciding what purpose or goal exists. After all CRs were assigned a goal, a Goal Ontology was generated, which can be found in Figure 5.

## Step 4: Operator, Value, and Boolean Ontology Creation

The only information that had not yet been derived out of each CR thus far in this methodology were values, operators, and Boolean values. These three pieces of information were needed to capture the fine details and if-statements imbedded in each CR. In this ontology, values represent any type of measurement or a reading that corresponds directly to a number i.e., set speed, low-pressure limit, set altitude, etc. Operators and Boolean values were necessary to include in order to portray events or comparisons visible in each CR. Again, these entities were determined for each CR following the prior process of first creating Table 11, then creating separate ontologies in

Protégé Owl for each category, which can be seen in Figures 6-8. Not all possible operators were enumerated, only the ones observed in the original 20 CR subset.

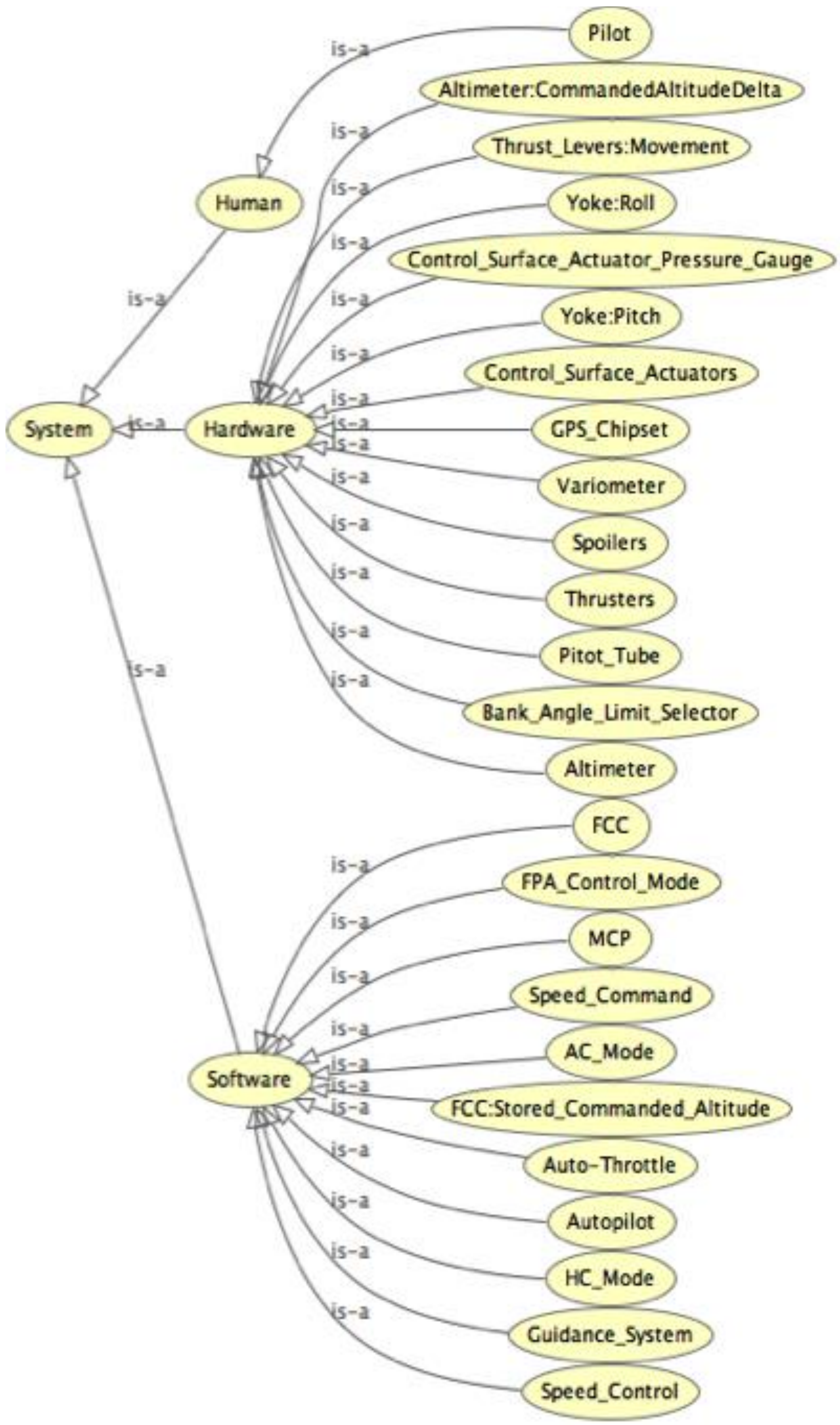


Figure 3: System Ontology

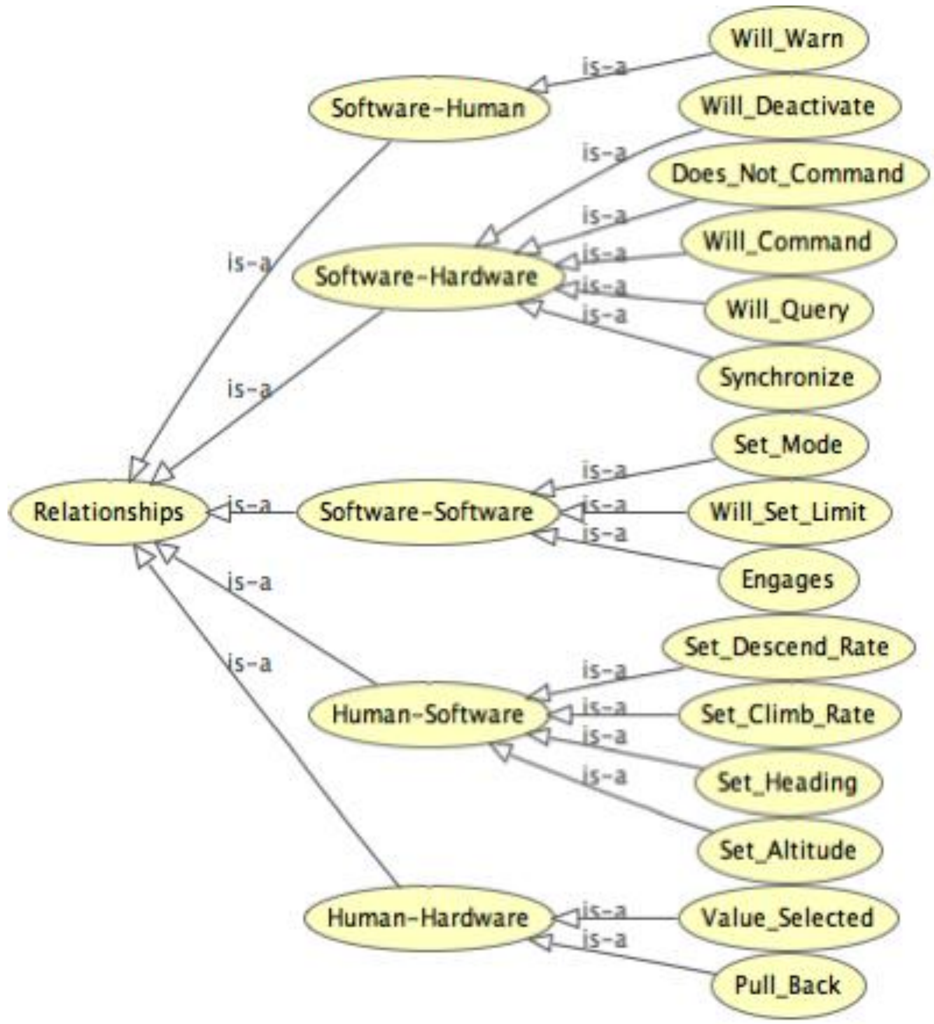


Figure 4: Relationship Ontology

## Step 5: Customer Requirement Template Creation

Next, a template for each CR was created using entities from each of the six created ontologies. This was accomplished by re-writing each CR using only ontology entities from the six generated ontologies and enough English words not lose any meaning or content from each CR. All 26 of these templates are labeled A Templates in Appendix B. The process used to analyze these six ontologies and 26 A Templates are captured in the results section of this paper.

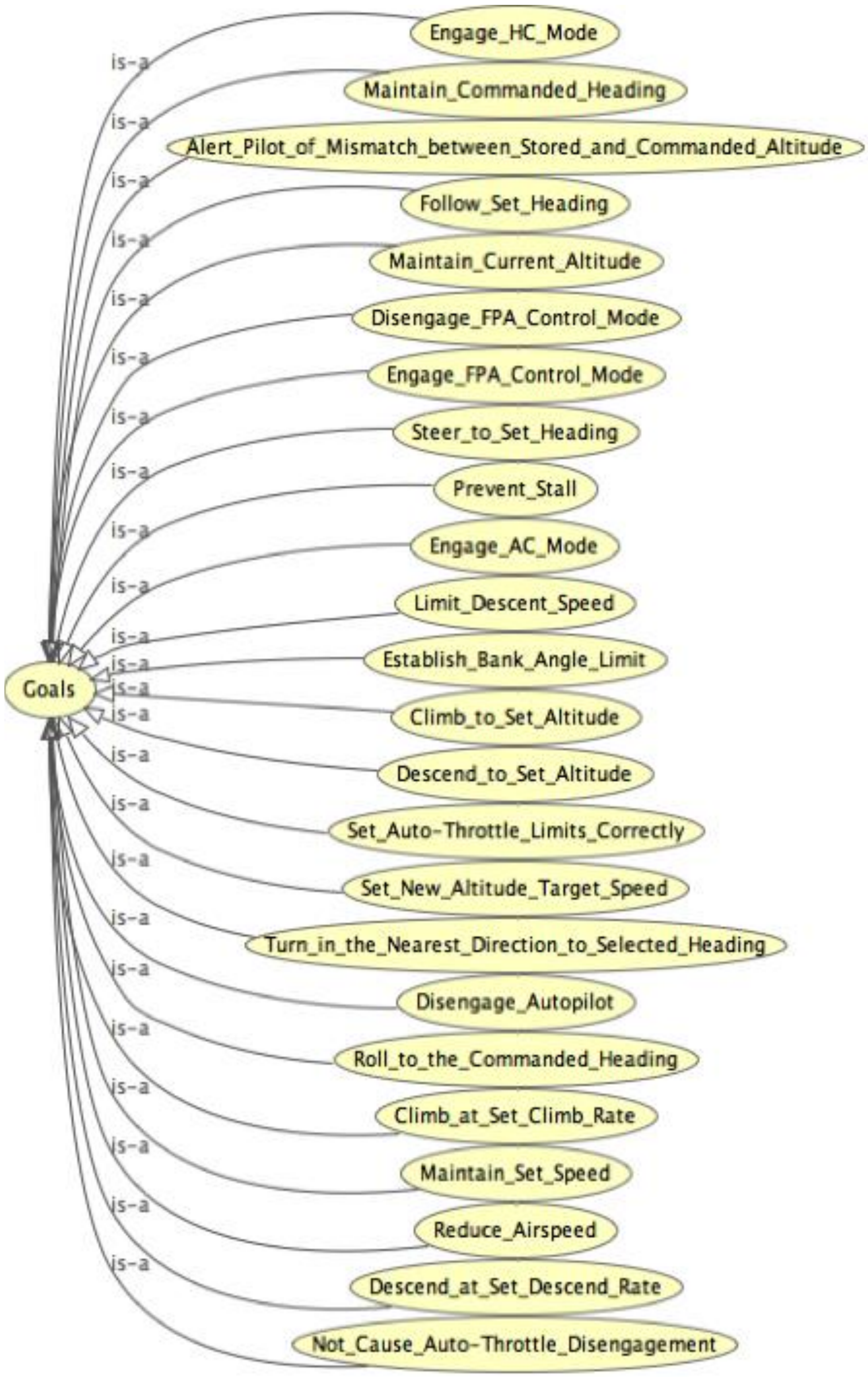


Figure 5: Goal Ontology

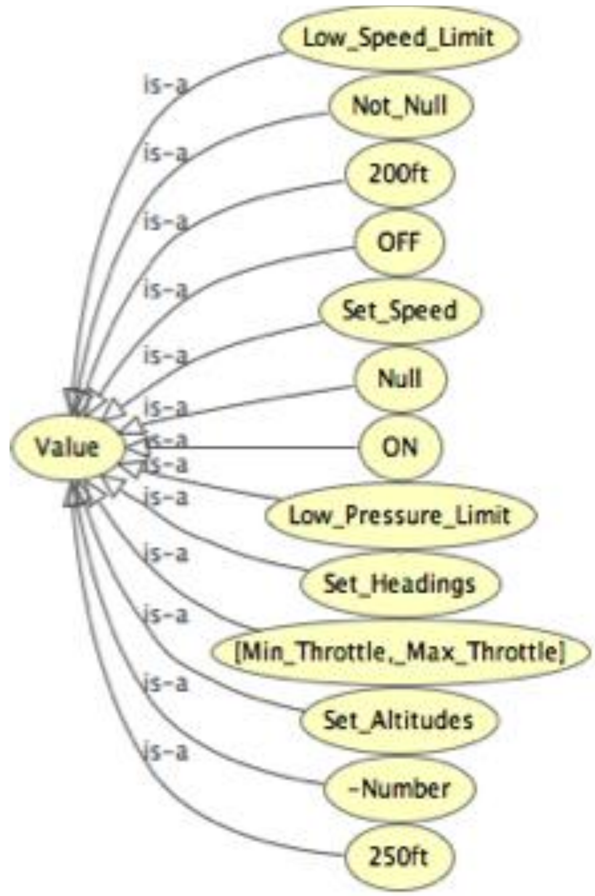


Figure 6: Value Ontology

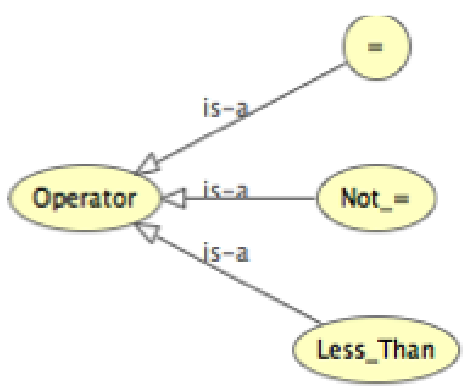


Figure 7: Operator Ontology

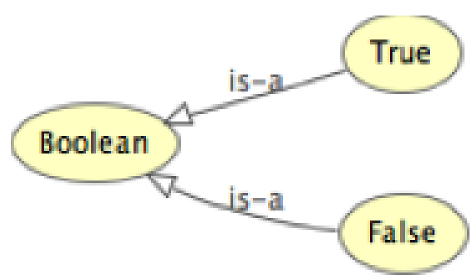


Figure 8: Boolean Ontology

# RESULTS

## Template Analysis

To further achieve the goal of improving the customer requirement formulation process, a template analysis was performed. The authors examined all 26 A Templates and noticed opportunity for further simplification and organization. Therefore, 26 new B Templates, also found in Appendix B, were created by eliminating the entity names in each template, but keeping the hierarchy ontology class name each entity was derived from, which again are: system, relationship, goal, operator, Boolean, goal. After this processing, it was clear that there were some underlying patterns that needed further investigation. The authors detected several repetitions with the order of the class names in relation to each other. Thus, Simplification Equations, found in Figure 9, were formed and plugged into each B Template to construct an even more simplified template, the C Template, for each CR, which can also be found in Appendix B. After these templates were studied, the authors discovered that there were four unique core C Templates that could represent all 26 CRs. These four unique templates will be referred to as the Final Customer Requirement Templates and can be found in Figure 10.

[Test] = [System] [Operation] [Value]

[Input Action] = [Relationship] [System]

[Response Action] = [System] [Relationship]

Figure 9: Simplification Equations

| |
|---|
| 1. If [Input Action] then [Response Action] in order for [Response Action] in order to [Goal] |
| 2. If [Input Action] then [Response Action] in order to [Goal] |
| 3. If Σ [Test] are/is [Boolean] then Σ [Response Action] in order to [Goal] |
| 4. [System] is limited to [Value] in order to [Goal] |

Figure 10: Final Customer Requirement Templates

## Customer Requirement Categorization

After splitting up each CR into groups corresponding to each of the four Final Customer Requirement Templates that they followed, a common theme emerged in each group. The authors agreed that there were three main types of CRs present: Monitor, Command, and Constraint. Monitor CRs require no human action or input. These CRs require a sensor or instrumentation to be constantly checking or monitoring the situation at hand. Command CRs require an input action from the pilot in order to be performed. Constraint CRs deal with limitations or constraints imposed on the physical conditions or system operation. There were 18 Monitor, 7 Command, and 1 Constraint present in this research, which can be broken down in more detail in Table 12.

## Customer Requirement Formulation Ontology

In order to create the final Customer Requirement Formulation Ontology, all six of the created ontologies were consolidated using Protégé Owl. Since this ontology is too large and complex to portray in a single diagram, Figure 11 depicts a condensed version showing how the six super classes and eight sub-classes are connected. This figure does not depict all of the eighty-two separate entities generated. The significance and use of this powerful ontology will be discussed in the next section.

Table 12: Customer Requirement Types

| | Monitor | Command | Constraint |
|---|---|---|---|
| **CR #** | 1, 3, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25 | 2, 4, 5, 6, 7, 9, 20 | 26 |
| **Total** | 18 | 7 | 1 |
| **Final CR Template Used** | 1, 2 | 3 | 4 |

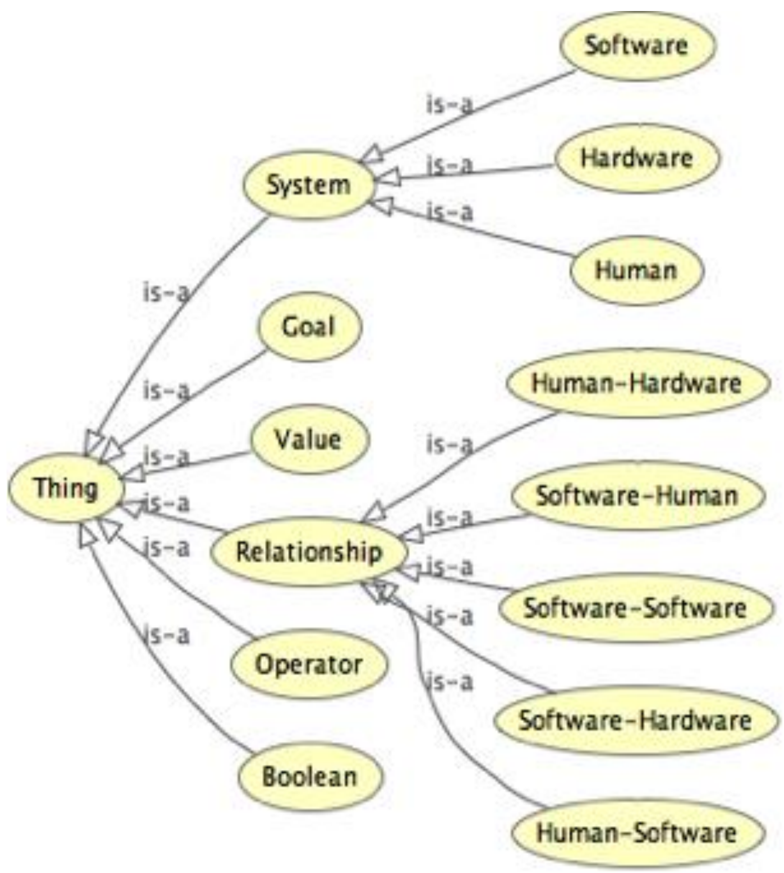


Figure 11: Customer Requirement Formulation Ontology

# DISCUSSION

## Application

There are many possibilities of how the NASA Ames Research Center could combine the findings of this research with their current customer requirement formulation tools. The use of the recently created Final Customer Requirement Templates and Customer Requirement Formulation Ontology is not a stand-alone proposition. These two resources will need to be supported by other tools and wizards in order to be used effectively. Because of proprietary reasons, the authors are not familiar with all the current capacities of NASA's tools and wizards, so the following scenario is the authors' best assumption on how this research might be utilized by NASA engineers.

When an engineer wants to create a CR for a target system, the engineer would open a wizard on their computer that would ask what type of CR the engineer is wanting to write, either: Command, Monitor, or Constraint. According to the answer, the wizard would then find and bring up the correct Final Customer Requirement Template for the engineer to fill out. Each template includes placeholders that would need to be filled in according to the content and meaning of the desired CR from pre-populated drop-down menus. The drop-down menu choices would be populated using corresponding entities from the Customer Requirement Formulation Ontology. Once all of the placeholders are filled in, the engineer would save the template, therefore creating a documented CR for the target system. For an example of what this process might look like, Figure 12 depicts the CR with placeholder text, drop-down menus and the final tabulated CR for an engineer attempting to capture the following command known as CR #2 in this study:

"The guidance system shall be able to automatically deploy spoilers when a significant reduction in airspeed is requested by the pilot."

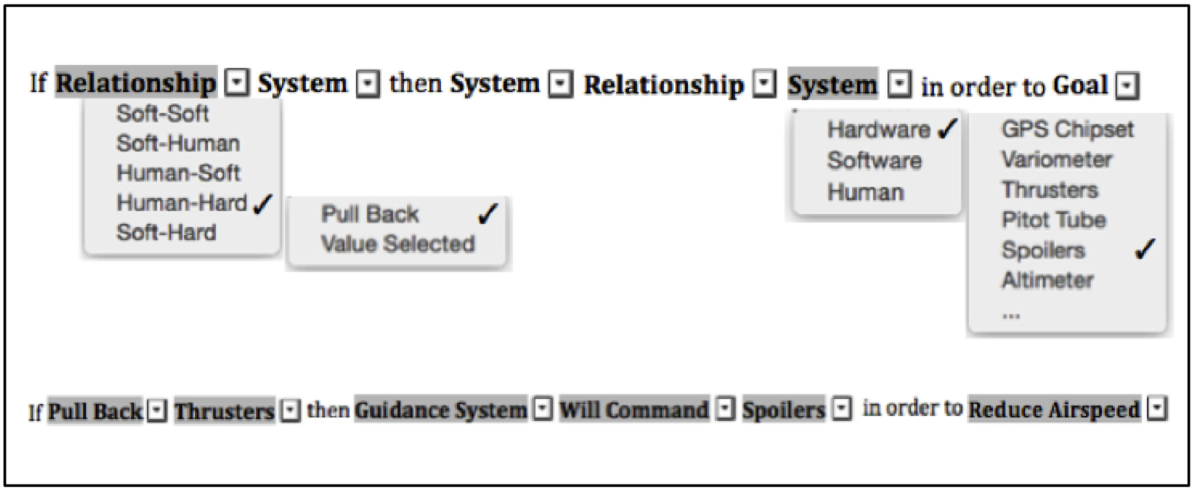


Figure 12: Example Template

## Benefits

There are many clear advantages in having CRs written in this template form. The main benefit is the elimination of ambiguity that is commonly seen in the CR writing phase of the design process, while still capturing all the systems and relationships that are involved in each CR. This has been achieved by expressing each CR with entities that are distinct and excluding unnecessary natural language jargon that can lead to misrepresentation and multiple interpretations. There are only ten natural language words that are used in all four Final Customer Requirement Templates, which include: is, and, in, order, to, for, then, if, limited, and are. All of these words are prepositions, conjunctions, or linking verbs, and do not cause any alteration of the meaning.

Simplifying the CR writing phase without disrupting the integrity of the content is a challenging task that this research attempts to alleviate.

Benefits regarding organization are evident with the application of this research. All 26 CRs that were examined in this research were able to be represented using only four templates. Engineers could query entity names or CR types against all CRs, which would increase productivity and organization. For example, an engineer could request a list of all CRs that involve thrusters or include a human. By standardizing the CR formulation process, future mistakes seen later in the design process due to ambiguity or disorganization can be reduced, which can save time and money. Overall, these practical tools used in this process can be applied across different engineering disciplines in order to create consistent, repeatable CRs.

## Teaching and Education

The findings of this research project can also impact the education sector. Collecting and interpreting CRs from the stakeholders of a product correctly is covered in many university engineering design classes. This is an important phase of the design process to grasp and implement because ambiguity in this step may lead to expensive and timely problems later in the design process. While professors and design texts cover this step in lectures, they could use the tools developed in this research paper to educate upcoming engineers on how to correctly construct, organize, and work with customer requirements minimizing ambiguity and confusion. The methodology and templates presented in this paper will not just impact current engineers, but future engineers as well.

## Limitations and Difficulties

There were some limitations that the authors would like to address regarding about this research project. First off, both authors are not subject matter experts in the guidance, navigation and control systems and sub-systems of the TCM; therefore their allocating of entities from the twenty provided CRs may contain some technical inaccuracies. However, the authors were not as concerned with this issue because the main focus of this project was to develop a usable methodology that encompasses innovative concepts and steps that can be followed easily. The study sponsor, NASA Ames Research Center, will likely seek subject matter experts to refine the entity selection for this ontology, however formulation of the high level ontological categories and the relationships between them coupled with a CR template was well received. Another difficulty the authors faced was the absence of already created ontology for customer requirement formulation use. Despite these difficulties, the authors feel confident that this research will be useful to engineers, students, teachers, and stakeholders.

# CONCLUSIONS/FUTURE WORK

The development of correctly written customer requirements is essential in producing design solutions that satisfy the end user. Because these requirements are carried throughout the entire design process, it is crucial to grasp the meaning and context as early as possible in the process. However, ambiguity and inconsistence may creep into these statements due to the use of natural language by designers. This problem needs to be resolved in order to save time, money, and accuracy. This research paper helps fill this gap in customer requirement engineering research by the construction of two resources: Final Customer Requirement Templates and Customer Requirement Formulation Ontology. These tools were derived by examining 20 guidance and control customer requirements provided by NASA Ames Research Center. The use of these aides will help address these customer requirement writing problems.

Since only 20 requirements were inspected during this paper, expansion and testing through future work persists. If more guidance control requirements are given, a verification test could be performed to see if each new requirement can be represented well using the two resources. After this process becomes robust, computational techniques to create automated customer requirements may be explored.

As mentioned earlier, the results of this research are not stand-alone. Both the templates and ontology created needs to be integrated with other software and wizard technologies that are in current use. NASA Ames Research Center or any engineer who is interested in improving the requirement engineering space may implement this work. An important aspect of the ontology development was establishing the type of relationships that exist between different categories. With this established, domain experts can define the ontology to be more exhaustive. The basic relationships are expected to have applicability beyond the TCM, aerospace field, and mechanical engineering field. Any complex system that deals with human-software-hardware interactions will likely have these relationships (at the high level). Thus, the templates would also have validity beyond the explored example. This work lays a foundation that should assist all design engineers to write thorough and robust customer requirements through the use of the templates and their supporting ontologies.

## *GENERAL CONCLUSION*

An ontology and taxonomy were successfully formulated during this research. Below are the conclusions from each paper that address the demands for a classification system that were outlined in the general introduction section for each application area.

Design Education

- For mechanical engineering courses, the number of theory classes could be reduced in a manner beneficial to the students with the adoption of more applied theory coursework. Currently non-existent studio based components could be effective in turning the heavy book problem based theory courses into applied theory courses where the students are more actively engaged in their learning.
- Studio classrooms, cooperative learning environments, and SCALE UP classrooms are setting techniques that take advantage of the amount of available technology, tutorials, and activities for students.
- A taxonomy categorizing class and setting type was created using a frequency analysis toolkit in Python along with a theory-driven approach. Then, it was populated using online course descriptions of five different design disciplines from five different design heavy universities.

Aerospace Complex Systems

- A methodology was created to uncover the existing relationships between the hardware and software levels found in aerospace CRs to create an appropriate ontology of systems and relationships that support compositional verification techniques.

- By combining the usefulness of the developed Customer Requirement Formulation Ontology and Final Customer Requirement Template, ambiguity, confusion, and uncertainty has been minimized from the CR formulation process.
- A categorizations scheme was devised in order to organize and store aerospace complex systems CRs better.

Similar and different approaches were used in the construction of the ontology and taxonomy. Top-down, inspirational, deductive, and local techniques were all utilized in the creation of each, however the use of a research corpus method was distinct to the taxonomy created for the design education application area. Along with these already proven techniques, the author created new, unique methodologies to help create these classification systems. All of the utilized approaches can be examined in more detail in the background and methods sections for each conference paper presented above.

Overall, these information organization resources are tied together because of the audience for which they are intended – the customer. A student concerned about his or her engineering education and customer requirement statements shares a common theme in that each setting puts the priority on end user. Implementing a method to better generate CRs and educating current mechanical engineering students with better classroom techniques for design directly benefit the end user.

In the end, the use of ontologies to describe the customer requirement space or to populate customer requirement templates gives the designer greater insight into the customer or user they are attempting to satisfy with a product or process. The research presented in this thesis adds to the field of engineering design by outlining formal methods that will remove ambiguity from the customer requirement gathering phase of design.

Additional work in both application areas is possible. Future refinement of the taxonomy by examining additional courses and using different criteria may be useful to better understand the design education field. The ontology mythology created to support CR generation in the aerospace field can be applicable to other complex systems because the steps followed are not domain specific. It is clear with this research that the benefits of ontologies and taxonomies can be applied to support repeatable and valid customer requirement generation.

## *BIBLIOGRAPHY*

[1] Chandrasekaran, B., and Josephson, J., and Benjamins, V. (1999). What Are Ontolgoies, and Why Do We Need Them? IEEE Intelligent Systems. 1094-7167/99.

[2] Lamar, C., (2009). Linguistic Analysis of Natural Language Engineering Requirements. All Thesis. Paper 671

[3] Mountain, J.R. The Decline of Design Across the Curriculum. in ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2008. American Society of Mechanical Engineers.

[4] Kazerounian, K. and Foley, S. Barriers To Creativity in Engineering Education: A Study of Instructors and Students Perceptions. Journal of Mechanical Design. 2007. 129(7): p. 761-768.

[5] Sauder, J. and Y. Jin. Training the Participatory Renaissance Man: Past Creative Experiences and Collaborative Design. in ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2013. American Society of Mechanical Engineers.

[6] Williams, C.B., et al. Exploring spatial reasoning ability and design cognition in undergraduate engineering students. in ASME 2010 international design engineering technical conferences and computers and information in engineering conference. 2010. American Society of Mechanical Engineers.

[7] Williams, C.B., et al. Exploring the effect of design education on the design cognition of mechanical engineering students. in ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2011. American Society of Mechanical Engineers.

[8] Lee, Y.S., J. Gero, and C.B. Williams. Comparing the Design Cognition of Two Engineering Majors: A Measurement-Based Approach. in ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2012. American Society of Mechanical Engineers.

[9] Williams, C.B., et al. Examining the effect of design education on the design cognition: Measurements from protocol studies. in Frontiers in Education Conference (FIE), 2012. 2012. IEEE.

[10] Roozenburg, N. and N. Cross, Models of the design process: integrating across the disciplines. Design Studies, 1991. **12**(4): p. 215-220.

[11] Otto, K. and K. Wood, Product Design: Techniques in Reverse Engineering, Systematic Design, and New Product Development. 2001, New York: Prentice-Hall.

[12] Ulrich, K. and E. SD, *Product Design and Development*. 4th ed. 2008, New York, New York: McGraw-Hill/Irwin.

[13] Ullman, D.G., *The Mechanical Design Process*. 4th ed. 2010, Boston: McGraw-Hill.

[14] Findeli, A., Rethinking design education for the 21st century: Theoretical, methodological, and ethical discussion. Design issues, 2001. **17**(1): p. 5-17.

[15] Ito, T. and A.H. Slocum. Teaching creative engineering: education in Japan and the USA. in ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2008. American Society of Mechanical Engineers.

[16] Smith, W.F. A Pillar of Mechanical Engineering Design Education in Australia – 25 Years of the Warman Design and Build Competition. in ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2013. American Society of Mechanical Engineers.
[17] Hodges, T.M. and G.A. Sullivan. Does an Experiential Based Project Improve Design Education? in ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2013. American Society of Mechanical Engineers.
[18] McKay, A., A. de Pennington, and J. Giard. Integrating design knowledge in educational curricula: reflections from the development of a multi-disciplinary programme in Product Design. in ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2013. American Society of Mechanical Engineers.
[19] Simpson, T.W., et al. Navigating the barriers to interdisciplinary design education: Lessons learned from the NSF design workshop series. in ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2010. American Society of Mechanical Engineers.
[20] Telenko, C., et al. Designettes: New approaches to multidisciplinary engineering design education. in ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2014. American Society of Mechanical Engineers.
[21] Watkins, S.M., Using the design process to teach functional apparel design. Clothing and Textiles Research Journal, 1988. **7**(1): p. 10-14.
[22] Regan, C.L., D.H. Kincade, and G. Sheldon, Applicability of the engineering design process theory in the apparel design process. Clothing and Textiles Research Journal, 1998. **16**(1): p. 36-46.
[23] Wilson, J.M. and W.C. Jennings, Studio courses: How information technology is changing the way we teach, on campus and off. Proceedings of the IEEE, 2000. **88**(1): p. 72-80.
[24] Little, P. and M. Cardenas, Use of "studio" methods in the introductory engineering design curriculum. Journal of Engineering Education, 2001. **90**(3): p. 309-318.
[25] Gross, M.D. and E.Y.-L. Do. The design studio approach: Learning design in architecture education. in Atlanta: Paper presented at the Design Education Workshop, EduTech/NSF, College of Computing, Georgia Institute of Technology. 1997.
[26] Sheppard, S., et al., *Examples of freshman design education.* International Journal of Engineering Education, 1997. **13**(4): p. 248-261.
[27] Sowa, J.F., *Top-level ontological categories.* International journal of human-computer studies, 1995. **43**(5): p. 669-685.
[28] Roy, N. and C. Hafner, The state of the art in ontology design. AI Magazine, 1997. **18**(3): p. 53-74.
[29] Uschold, M. and M. Gruninger, Ontologies: Principles, methods and applications. Knowledge engineering review, 1996. **11**(2): p. 93-136.
[30] Holsapple, C.W. and K.D. Joshi, A collaborative approach to ontology design. Communications of the ACM, 2002. **45**(2): p. 42-47.

[31] Dahlgren, K., Naive semantics for natural language understanding. 1988: Springer.
[32] van der Vet, P.E., P.-H. Speel, and N.J. Mars. The Plinius ontology of ceramic materials. in Eleventh European Conference on Artificial Intelligence (ECAI'94) Workshop on Comparison of Implemented Ontologies. 1994. Citeseer.
[33] Bird, S. NLTK: the natural language toolkit. in Proceedings of the COLING/ACL on Interactive presentation sessions. 2006. Association for Computational Linguistics.
[34] Cohen, J., *A coefficient of agreement for nominal scales.* Educational and Psychological Measurement, 1960.
[35] Landis, J.R. and G.G. Koch, The measurement of observer agreement for categorical data. biometrics, 1977: p. 159-174.
[36] Beichner, R.J., et al., The student-centered activities for large enrollment undergraduate programs (SCALE-UP) project. Research-based reform of university physics, 2007. **1**(1): p. 2-39.
[37] Little, A.J. and B.A. León de la Barra, Attracting girls to science, engineering and technology: an Australian perspective. European Journal of Engineering Education, 2009. **34**(5): p. 439-445.
[38] Ivey, S., et al., Attracting Students to Transportation Engineering: Gender Differences and Implications of Student Perceptions of Transportation Engineering Careers. Transportation Research Record: Journal of the Transportation Research Board, 2012(2320): p. 90-96.
[39] Brat, G., Bushnell, D., Davies, M., Giannakopoulou, D., Howar, F., Kahsai, T., Verifying the Safety of a Flight-Critical System. Technical report, NASA, Ames Reseach Center, Moffett Field, CA
[40] R. M. Hueschen. Development of the Transport Class Model (TCM) aircraft simulation from a sub-scale Generic Transport Motel (GTM) simulation. Technical report, NASA, Langley Research Center, Hampton, VA, August 2011
[41] Gruber, T., Toward principles for the design of ontologies used for knowledge sharing. Int. J. Human-Computer Studies (1995) **43**, 907-928
[42] Gruber, T., A Translation Approach to Portable Ontology Specifications. Knowledge Systems Laboratory Technical Report KSL 92-71
[43] Noy, N., and McGuinness, D., Ontology Development 101: A Guide to Create Your First Ontology. Technical Report. Stanford University, Stanford, CA
[44] Knublauch, H., Horridge, M., Musen, M., Rector, A., Stevens, R., Drummond, N., and Lord, P., The Protégé Owl Experience. Technical Report
[45] Knublauch, H., Fergerson, R., Noy, N., and Musen, M., The Protégé Owl Plugin: An Open Development Environment for Semantic Web Applications. The Semantic Web- ISWC (2004) **3298**, 229-243
[46] Knublach, H., Ontology-Driven Software Development in the Context of the Semantic Web: An Example Scenario with Protégé/Owl. Technical paper. Stanford Medical Informatics, Stanford University
[47] Hooks, I., Writing Good Requirements (A Requirements Working Group Information Report). Third International Symposium of the NCOSE – Volume 2, 1993.
[48] Robertson, S., and Robertson, J., Mastering the Requirement Process: Getting Requirements Right. 3rd ed. Ch. 2 pp. 20-22

# ***APPENDICES***

# APPENDIX A: CUSTOMER REQUIREMENT & ENTITY TABLES

Table 6: List of Original Customer Requirements

| CR # | Customer Requirement |
|---|---|
| 1 | The guidance system shall be capable of maintaining a steady speed in the normal flight envelope. |
| 2 | The guidance system shall be capable of steering to and following a specified heading. |
| 3 | The guidance system shall be capable of climbing at a defined rate, to be limited by minimum and maximum engine performance and airspeeds. |
| 4 | The guidance system shall be capable of descending at a defined rate, to be limited by minimum and maximum engine performance and airspeeds. |
| 5 | The guidance system shall be capable of climbing at a specified rate to a specified altitude, to be limited by maximum engine performance for a set airspeed. |
| 6 | The guidance system shall be capable of descending at a specified rate to a specified altitude, to be limited by maximum engine performance for a set airspeed. |
| 7 | The guidance system shall be able to automatically deploy spoilers to limit speed in a descent, or when a significant reduction in airspeed is requested by the pilot, deactivating at low speed. |
| 8 | The Altitude Control shall engage when the Altitude Control mode is selected and when the FPA Control mode is not selected, and when there is no manual pitch or manual roll command from the stick. |
| 9 | The FPA Control shall engage when the FPA Control mode is selected and when there is no manual pitch or manual roll command from the stick. |
| 10 | If any control surface actuator loses hydraulic pressure, the Autopilot shall disengage. |
| 11 | If the Altitude Control is engaged, once the plane is within 250 ft. of the commanded altitude, the plane will remain within 250 ft. of the commanded altitude. |
| 12 | If the FPA Control and the Altitude Control are both selected, the FPA Control will disengage and the Altitude Control will engage once the plane is within 200 ft. of the commanded altitude. |
| 13 | The Heading Control shall engage when the Heading Control mode is selected and when there is no manual pitch or manual roll command from the stick. |

Table 6: List of Original Customer Requirements (Continued)

| | |
|---|---|
| 14 | If the Altitude Control is engaged with no active Speed Control, the Speed Control shall engage and the speed command shall synchronize to the current speed, which shall become the new altitude's target speed. |
| 15 | The bank angle limit is established by the Bank Angle Limit Selector. |
| 16 | The Heading Control mode, when selected, sends roll commands to turn to and maintain the commanded heading. |
| 17 | When the Heading Control mode is engaged, roll commands are given to turn in the nearest direction to the selected heading. |
| 18 | Manually positioning the thrust levers does not cause Auto-Throttle disengagement. |
| 19 | The FCCs shall issue a warning when the commanded altitude disagrees with the stored commanded altitude stored in the FCCs. |
| 20 | The Auto-Throttle will be limited by the maximum throttle, and the minimum throttle. |

Table 7: List of Edited Customer Requirements

| New CR # | Original CR# | Customer Requirement |
|---|---|---|
| 1 | 1 | The guidance system shall be capable of maintaining a steady speed in the normal flight envelope. * |
| 2 | 2 | The guidance system shall be capable of steering to a specified heading. |
| 3 | 2 | The guidance system shall be capable of following a specified heading. |
| 4 | 3 | The guidance system shall be capable of climbing at a defined rate. |
| 5 | 4 | The guidance system shall be capable of descending at a defined rate. |
| 6 | 5 | The guidance system shall be capable of climbing to a specified altitude. |
| 7 | 6 | The guidance system shall be capable of descending to a specified altitude. |
| 8 | 7 | The guidance system shall be able to automatically deploy spoilers to limit speed in a descent. |
| 9 | 7 | The guidance system shall be able to automatically deploy spoilers when a significant reduction in airspeed is requested by the pilot. |
| 10 | 7 | The guidance system shall be able to automatically deactivate spoilers at low speed. |
| 11 | 8 | The Altitude Control shall engage when the Altitude Control mode is selected and when the FPA Control mode is not selected. |
| 12 | 8 | The Altitude Control shall engage when there is no manual pitch or manual roll command from the stick. |
| 13 | 9 | The FPA Control shall engage when the FPA Control mode is selected and when there is no manual pitch or manual roll command from the stick. * |
| 14 | 10 | The Autopilot shall disengage if any control surface actuator loses hydraulic pressure. |
| 15 | 11 | The plane will remain within 250 ft. of the commanded altitude if the Altitude Control is engaged and the plane is within 250 ft. of the commanded altitude. |

Table 7: List of Edited Customer Requirements (Continued)

| | | |
|---|---|---|
| 16 | 12 | The FPA Control will disengage if the FPA Control and the Altitude Control are both selected. |
| 17 | 12 | The Altitude Control will engage if the plane is within 200 ft. of the commanded altitude and if the FPA Control and the Altitude Control are both selected. |
| 18 | 13 | The Heading Control shall engage when the Heading Control mode is selected and when there is no manual pitch or manual roll command from the stick. * |
| 19 | 14 | The Speed Control shall engage and the speed command shall synchronize to the current speed if the Altitude Control is engaged with no active Speed Control, which shall become the new altitude's target speed. |
| 20 | 15 | The bank angle limit is established by the Bank Angle Limit Selector. * |
| 21 | 16 | The Heading Control mode, when selected, sends roll commands to turn to the commanded heading. |
| 22 | 16 | The Heading Control mode, when selected, sends roll commands to maintain the commanded heading. |
| 23 | 17 | When the Heading Control mode is engaged, roll commands are given to turn in the nearest direction to the selected heading. * |
| 24 | 18 | Manually positioning the thrust levers will not cause Auto-Throttle disengagement. * |
| 25 | 19 | The FCCs shall issue a warning when the commanded altitude disagrees with the stored commanded altitude stored in the FCCs.* |
| 26 | 20 | The Auto-Throttle will be limited by the maximum throttle and the minimum throttle. * |
| | | * - No alteration from original CR |

Table 8: System Entities

| CR # | Hardware | Software | Human |
|---|---|---|---|
| 1 | Control Surface Actuators, Thrusters | Speed Control, Guidance System | |
| 2 | GPS Chipset, Control Surface Actuators Thrusters | MCP, Guidance System | |
| 3 | Control Surface Actuators, Thrusters | HC Mode, Guidance System | |
| 4 | Variometer, Control Surface Actuators, Thrusters | MCP, Guidance System | |
| 5 | Variometer, Control Surface Actuators, Thrusters | MCP, Guidance System | |
| 6 | Altimeter, Control Surface Actuators, Thrusters | MCP, Guidance System | |
| 7 | Altimeter, Control Surface Actuators, Thrusters | MCP, Guidance System | |
| 8 | Variometer, Spoilers | Guidance System | |
| 9 | Thrusters, Spoilers | Guidance System | |
| 10 | Pitot Tube, Spoilers | Guidance System | |
| 11 | | AC Mode, FPA Control Mode, Guidance System | |
| 12 | Yoke:Pitch, Yoke:Roll | Guidance System, AC Mode | |
| 13 | Yoke:Pitch, Yoke:Roll | FPA Control Mode, Guidance System | |
| 14 | Control Surface Actuator Pressure Gauge | Guidance System, Autopilot | |
| 15 | Altimeter:Commanded Altitude Δ, Control Surface Actuators, Thrusters | AC Mode | |
| 16 | | FPA Control Mode, Guidance System, AC Mode | |
| 17 | Altimeter:Commanded | FPA Control Mode, AC | |

Table 8: System Entities (Continued)

| | | | |
|---|---|---|---|
| | Altitude Δ | Mode, Guidance System | |
| 18 | Yoke:Pitch, Yoke:Roll | HC Mode, Guidance System | |
| 19 | Pitot Tube | AC Mode, Speed Control, Guidance System, Speed Command | |
| 20 | Bank Angle Limit Selector | Guidance System, HC Mode | |
| 21 | Control Surface Actuators | HC Mode | |
| 22 | Control Surface Actuators | HC Mode | |
| 23 | Control Surface Actuators | HC Mode | |
| 24 | Thrust Levers:Movement | Auto-Throttle, Guidance System | |
| 25 | | FCC:Stored Commanded Altitude, FCC | Pilot |
| 26 | | Auto-Throttle | |

Table 9: Relationship Entities

| CR # | Software-Hardware | Software-Software | Human-Software | Human-Hardware | Software-Human |
|---|---|---|---|---|---|
| 1 | Will Command | | | | |
| 2 | Will Query, Will Command | | Set Heading | | |
| 3 | Will Command | | | | |
| 4 | Will Query, Will Command | | Set Climb Rate | | |
| 5 | Will Query, Will Command | | Set Descend Rate | | |
| 6 | Will Query, Will Command | | Set Altitude | | |
| 7 | Will Query, Will Command | | Set Altitude | | |
| 8 | Will Command | | | | |
| 9 | Will Command | | | Pull Back | |
| 10 | Will Deactivate | | | | |
| 11 | | Set Mode | | | |
| 12 | | Set Mode | | | |
| 13 | | Set Mode | | | |
| 14 | | Set Mode | | | |
| 15 | Does Not Command | | | | |
| 16 | | Set Mode | | | |

Table 9: Relationship Entities (Continued)

| | | | | | |
|---|---|---|---|---|---|
| 17 | | Set Mode | | | |
| 18 | | Set Mode | | | |
| 19 | Synchronize | Set Mode | | | |
| 20 | | Will Set Limit | | Value Selected | |
| 21 | Will Command | | | | |
| 22 | Will Command | | | | |
| 23 | Will Command | | | | |
| 24 | | Engages | | | |
| 25 | | | | | Will Warn |
| 26 | | | | | |

Table 10: Goal Entities

| CR # | Goal |
|---|---|
| 1 | Maintain Set Speed |
| 2 | Steer to Set Heading |
| 3 | Follow Set Heading |
| 4 | Climb at Set Climb Rate |
| 5 | Descend at Set Descend Rate |
| 6 | Climb to Set Altitude |
| 7 | Descend to Set Altitude |
| 8 | Limit Descent Speed |
| 9 | Reduce Airspeed |
| 10 | Prevent Stall |
| 11 | Engage AC Mode |
| 12 | Engage AC Mode |
| 13 | Engage FPA Control Mode |
| 14 | Disengage Autopilot |
| 15 | Maintain Current Altitude |
| 16 | Disengage FPA Control Mode |
| 17 | Engage AC Mode |
| 18 | Engage HC Mode |
| 19 | Set New Altitude Target Speed |
| 20 | Establish Bank Angle Limit |
| 21 | Roll to the Commanded Heading |
| 22 | Maintain Commanded Heading |
| 23 | Turn in the Nearest Direction to Selected Heading |
| 24 | Not Cause Auto-Throttle Disengagement |
| 25 | Alert Pilot of Mismatch between Stored and Commanded Altitude |
| 26 | Set Auto-Throttle Limits Correctly |

Table 11: Operator, Value, and Boolean Entities

| CR # | Operator | Value | Boolean |
|---|---|---|---|
| 1 | = | Set Speed | True |
| 2 | | | |
| 3 | = | Set Heading | True |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | = | - # | True |
| 9 | | | |
| 10 | < | Low Speed Limit | True |
| 11 | = | ON, OFF | True |
| 12 | = | Null | True |
| 13 | = | ON, Null | True |
| 14 | < | Low Pressure Limit | True |
| 15 | < = | ON, 250ft | True |
| 16 | = | ON | True |
| 17 | < = | ON, 200ft | True |
| 18 | = | ON, Null | True |
| 19 | = | ON, OFF | True |
| 20 | | | |
| 21 | = | ON | True |
| 22 | = | ON | True |
| 23 | = | ON | True |
| 24 | = | Not Null | True |
| 25 | not = | Set Altitude | True |
| 26 | | [Min Throttle, Max Throttle] | |

# APPENDIX B: CUSTOMER REQUIREMENT TEMPLATES A-D

1. The guidance system shall be capable of maintaining a steady speed in the normal flight envelope.

A: If [**Software**: Speed Control] [**Operator**: = ] [**Value**: Set Speed] is [**Boolean**: True] then [**Software**: Guidance System] [**Soft-Hard**: Will Command] [**Hardware**: Control Surface Actuators, Thrusters] in order to [**Goal**: Maintain Set Speed].

B: If [**System**] [**Operator**] [**Value**] is [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] is [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____is____then____in order to____.

CR Type: Monitor

2. The guidance system shall be capable of steering to a specified heading.

A: If [**Human-Soft**: Set Heading] [**Software**: MCP] then [**Software**: Guidance System] [**Soft-Hard**: Will Query] [**Hardware**: GPS Chipset] in order for [**Software**: Guidance System] [**Soft-Hard**: Will Command] [**Hardware**: Control Surface Actuators, Thrusters] in order to [**Goal**: Steer to Set Heading].

B: If [**Relationship**] [**System**] then [**System**] [**Relationship**] [**System**] in order for [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Input Action**] then [**Response Action**] in order for [**Response Action**] in order to [**Goal**].

D: If____then____in order for____in order to____.

CR Type: Command

3. The guidance system shall be capable of following a specified heading.

A: If [**Software**: HC Mode] [**Operation**: = ] [**Value**: Set Heading] is [**Boolean**: True] then [**Software**: Guidance System] [**Soft-Hard**: Will Command] [**Hardware**: Control Surface Actuators, Thrusters] in order to [**Goal**: Follow Set Heading].

B: If [**System**] [**Operator**] [**Value**] is [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] is [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____is____then____in order to____.

CR Type: Monitor

4. The guidance system shall be capable of climbing at a defined rate.

A: If [**Human-Soft**: <u>Set Climb Rate</u>] [**Software**: <u>MCP</u>] then [**Software**: <u>Guidance System</u>] [**Soft-Hard**: <u>Will Query</u>] [**Hardware**: <u>Variometer</u>] in order for [**Software**: <u>Guidance System</u>] [**Soft-Hard**: <u>Will Command</u>] [**Hardware**: <u>Control Surface Actuators, Thrusters</u>] in order to [**Goal**: <u>Climb at Set Climb Rate</u>].

B: If [**Relationship**] [**System**] then [**System**] [**Relationship**] [**System**] in order for [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Input Action**] then [**Response Action**] in order for [**Response Action**] in order to [**Goal**].

D: If____then____in order for____in order to____.

CR Type: Command

5. The guidance system shall be capable of descending at a defined rate.

A: If [**Human-Soft**: <u>Set Descend Rate</u>] [**Software**: <u>MCP</u>] then [**Software**: <u>Guidance System</u>] [**Soft-Hard**: <u>Will Query</u>] [**Hardware**: <u>Variometer</u>] in order for [**Software**: <u>Guidance System</u>] [**Soft-Hard**: <u>Will Command</u>] [**Hardware**: <u>Control Surface Actuators, Thrusters</u>] in order to [**Goal**: <u>Descend at Set Descend Rate</u>].

B: If [**Relationship**] [**System**] then [**System**] [**Relationship**] [**System**] in order for [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Input Action**] then [**Response Action**] in order for [**Response Action**] in order to [**Goal**].

D: If____then____in order for____in order to____.

CR Type: Command

6. The guidance system shall be capable of climbing to a specified altitude.

A: If [**Human-Soft**: <u>Set Altitude</u>] [**Software**: <u>MCP</u>] then [**Software**: <u>Guidance System</u>] [**Soft-Hard**: <u>Will Query</u>] [**Hardware**: <u>Altimeter</u>] in order for [**Software**: <u>Guidance System</u>] [**Soft-Hard**: <u>Will Command</u>] [**Hardware**: <u>Control Surface Actuators, Thrusters</u>] in order to [**Goal**: <u>Climb to Set Altitude</u>].

B: If [**Relationship**] [**System**] then [**System**] [**Relationship**] [**System**] in order for [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Input Action**] then [**Response Action**] in order for [**Response Action**] in order to [**Goal**].

D: If____then____in order for____in order to____.

CR Type: Command

7. The guidance system shall be capable of descending to a specified altitude.

A: If [**Human-Soft**: <u>Set Altitude</u>] [**Software**: <u>MCP</u>] then [**Software**: <u>Guidance System</u>] [**Soft-Hard**: <u>Will Query</u>] [**Hardware**: <u>Altimeter</u>] in order for [**Software**: <u>Guidance System</u>] [**Soft-Hard**: <u>Will Command</u>] [**Hardware**: <u>Control Surface Actuators, Thrusters</u>] in order to [**Goal**: <u>Descend to Set Altitude</u>].

B: If [**Relationship**] [**System**] then [**System**] [**Relationship**] [**System**] in order for [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Input Action**] then [**Response Action**] in order for [**Response Action**] in order to [**Goal**].

D: If____then____in order for____in order to____.

CR Type: Command

8. The guidance system shall be able to automatically deploy spoilers to limit speed in a descent.

A: If [**Hardware**: <u>Variometer</u>] [**Operator**: = ] [**Value**: <u>-#</u> ] is [**Boolean**: <u>True</u>] then [**Software**: <u>Guidance System</u>] [**Soft-Hard**: <u>Will Command</u>] [**Hardware**: <u>Spoilers</u>] in order to [**Goal**: <u>Limit Descent Speed</u>].

B: If [**System**] [**Operator**] [**Value**] is [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] is [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____is____then____in order to____.

CR Type: Monitor

9. The guidance system shall be able to automatically deploy spoilers when a significant reduction in airspeed is requested by the pilot.

A: If [**Human-Hard**: Pull Back] [**Hardware**: Thrusters] then [**Software**: Guidance System] [**Soft-Hard**: Will Command] [**Hardware**: Spoilers] in order to [**Goal**: Reduce Airspeed].

B: If [**Relationship**] [**System**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Input Action**] then [**Response Action**] in order to [**Goal**].

D: If____then____ in order to____.

CR Type: Command

10. The guidance system shall be able to automatically deactivate spoilers at low speed.

A: If [**Hardware**: Pitot Tube] [**Operator**: ≤ ] [**Value**: Low Speed Limit] is [**Boolean**: True] then [**Software**: Guidance System] [**Soft-Hard**: Will Deactivate] [**Hardware**: Spoilers] in order to [**Goal**: Prevent Stall].

B: If [**System**] [**Operator**] [**Value**] is [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] is [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____is____then____in order to____.

CR Type: Monitor

11. The Altitude Control shall engage when the Altitude Control mode is selected and when the FPA Control mode is not selected.

A: If [**Software**: AC Mode] [**Operator**: = ] [**Value**: ON ] and [**Software**: FPA Control Mode] [**Operator**: = ] [**Value**: OFF ] are [**Boolean**: True] then [**Software**: Guidance System] [**Soft-Soft**: Set Mode] [**Software**: AC Mode] in order to [**Goal**: Engage AC Mode].

B: If [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] are [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] and [**Test**] are [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____and____are____then____in order to____.

CR Type: Monitor

12. The Altitude Control shall engage when there is no manual pitch or manual roll command from the stick.

A: If [**Hardware**: Yoke:Pitch] [**Operator**: = ] [**Value**: Null] and [**Hardware**: Yoke:Roll] [**Operator**: = ] [**Value**: Null] are [**Boolean**: True] then [**Software**: Guidance System] [**Soft-Soft**: Set Mode] [**Software**: AC Mode] in order to [**Goal**: Engage AC Mode].

B: If [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] are [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] and [**Test**] are [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____and____are____then____in order to____.

CR Type: Monitor

13. The FPA Control shall engage when the FPA Control mode is selected and when there is no manual pitch or manual roll command from the stick.

A: If [**Software**: FPA Control Mode] [**Operator**: = ] [**Value**: ON] and [**Hardware**: Yoke:Pitch] [**Operator**: = ] [**Value**: Null] and [**Hardware**: Yoke:Roll] [**Operator**: = ] [**Value**: Null] are [**Boolean**: True] then [**Software**: Guidance System] [**Soft-Soft**: Set Mode] [**Software**: FPA Control Mode] in order to [**Goal**: Engage FPA Control Mode].

B: If [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] are [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] and [**Test**] and [**Test**] are [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____ and ____and____are____then____in order to____.

CR Type: Monitor

14. The Autopilot shall disengage if any control surface actuator loses hydraulic pressure.

A: If [**Hardware**: Control Surface Actuator Pressure Gauge] [**Operator**: ≤ ] [**Value**: Low Pressure Limit] is [**Boolean**: True] then [**Software**: Guidance System] [**Soft-Soft**: Set Mode] [**Software**: Autopilot] in order to [**Goal**: Disengage Autopilot].

B: If [**System**] [**Operator**] [**Value**] is [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] is [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____ is____then____in order to____.

CR Type: Monitor

15. The plane will remain within 250 ft of the commanded altitude if the Altitude Control is engaged and the plane is within 250 ft of the commanded altitude.

A: If [**Software**: AC Mode] [**Operator**: = ] [**Value**: ON] and [**Hardware**: Altimeter:Commanded Altitude **Δ**] [**Operator**: ≤ ] [**Value**: 250ft] are [**Boolean**: True] then [**Software**: AC Mode] [**Soft-Hard**: Does Not Command] [**Hardware**: Control Surface Actuators, Thrusters] in order to [**Goal**: Maintain Current Altitude].

B: If [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] are [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] and [**Test**] are [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____ and ____are____then____in order to____.

CR Type: Monitor

16. The FPA Control will disengage if the FPA Control and the Altitude Control are both selected.

A: If [**Software**: FPA Control Mode] [**Operator**: = ] [**Value**: ON] and [**Software**: AC Mode] [**Operator**: = ] [**Value**: ON] are [**Boolean**: True] then [**Software**: Guidance System] [**Soft-Soft**: Set Mode] [**Software**: FPA Control Mode] in order to [**Goal**: Disengage FPA Control Mode].

B: If [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] are [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] and [**Test**] are [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____and____are____then____in order to____.

CR Type: Monitor

17. The Altitude Control will engage if the plane is within 200 ft of the commanded altitude and if the FPA Control and the Altitude Control are both selected.

A: If [**Software**: FPA Control Mode] [**Operator**: = ] [**Value**: ON] and [**Software**: AC Mode] [**Operator**: = ] [**Value**: ON] and [**Hardware**: Altimeter:Commanded Altitude Δ] [**Operator**: ≤ ] [**Value**: 200ft] are [**Boolean**: True] then [**Software**: Guidance System] [**Soft-Soft**: Set Mode] [**Software**: AC Mode] in order to [**Goal**: Engage AC Mode].

B: If [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] are [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] and [**Test**] and [**Test**] are [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____ and____and____are____then____in order to____.

CR Type: Monitor

18. The Heading Control shall engage when the Heading Control mode is selected and when there is no manual pitch or manual roll command from the stick.

A: If [**Software**: HC Mode] [**Operator**: = ] [**Value**: ON] and [**Hardware**: Yoke:Pitch] [**Operator**: = ] [**Value**: Null] and [**Hardware**: Yoke:Roll] [**Operator**: = ] [**Value**: Null] are [**Boolean**: True] then [**Software**: Guidance System] [**Soft-Soft**: Set Mode] [**Software**: HC Mode] in order to [**Goal**: Engage HC Mode].

B: If [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] are [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] and [**Test**] and [**Test**] are [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____ and ____and____are____then____in order to____.

CR Type: Monitor

19. The Speed Control shall engage and the speed command shall synchronize to the current speed if the Altitude Control is engaged with no active Speed Control, which shall become the new altitude's target speed.

A: If [**Software**: <u>AC Mode</u>] [**Operator**: <u>=</u> ] [**Value**: <u>ON</u>] and [**Software**: <u>Speed Control</u>] [**Operator**: <u>=</u> ] [**Value**: <u>OFF</u>] are [**Boolean**: <u>True</u>] then [**Software**: <u>Guidance System</u>] [**Soft-Soft**: <u>Set Mode</u>] [**Software**: <u>Speed Control</u>] and [**Software**: <u>Speed Command</u>] [**Soft-Hard**: <u>Synchronize</u>] [**Hardware**: <u>Pitot Tube</u>] in order to [**Goal**: <u>Set New Altitude Target Speed</u>].

B: If [**System**] [**Operator**] [**Value**] and [**System**] [**Operator**] [**Value**] are [**Boolean**] then [**System**] [**Relationship**] [**System**] and [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] and [**Test**] are [**Boolean**] then [**Response Action**] and [**Response Action**] in order to [**Goal**].

D: If____ and ____are____then____and____in order to____.

CR Type: Monitor

20. The bank angle limit is established by the Bank Angle Limit Selector.

A: If [**Human-Hard**: <u>Value Selected</u>] [**Hardware**: <u>Bank Angle Limit Selector</u>] then [**Software**: <u>Guidance System</u>] [**Soft-Soft**: <u>Will Set Limit</u>] [**Software**: <u>HC Mode</u>] in order to [**Goal**: <u>Establish Bank Angle Limit</u>].

B: If [**Relationship**] [**System**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Input Action**] then [**Response Action**] in order to [**Goal**].

D: If____then____in order to____.

CR Type: command

21. The Heading Control mode, when selected, sends roll commands to turn to the commanded heading.

A: If [**Software**: HC Mode] [**Operator**: = ] [**Value**: ON] is [**Boolean**: True] then [**Software**: HC Mode] [**Soft-Hard**: Will Command] [**Hardware**: Control Surface Actuators] in order to [**Goal**: Roll to the Commanded Heading].

B: If [**System**] [**Operator**] [**Value**] is [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] is [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____is____then____in order to____.

CR Type: Monitor

22. The Heading Control mode, when selected, sends roll commands to maintain the commanded heading.

A: If [**Software**: HC Mode] [**Operator**: = ] [**Value**: ON] is [**Boolean**: True] then [**Software**: HC Mode] [**Soft-Hard**: Will Command] [**Hardware**: Control Surface Actuators] in order to [**Goal**: Maintain Commanded Heading].

B: If [**System**] [**Operator**] [**Value**] is [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] is [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____is____then____in order to____.

CR Type: Monitor

23. When the Heading Control mode is engaged, roll commands are given to turn in the nearest direction to the selected heading.

A: If [**Software**: HC Mode] [**Operator**: = ] [**Value**: ON] is [**Boolean**: True] then [**Software**: HC Mode] [**Soft-Hard**: Will Command] [**Hardware**: Control Surface Actuators] in order to [**Goal**: Turn in the Nearest Direction to Selected Heading].

B: If [**System**] [**Operator**] [**Value**] is [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] is [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____is____then____in order to____.

CR Type: monitor

24. Manually positioning the thrust levers will not cause Auto-Throttle disengagement.

A: If [**Hardware**: Thrust Levers:Movement] [**Operation**: = ] [**Value**: Not Null] is [**Boolean**: True] then [**Software**: Guidance System ] [**Soft-Soft**: Engages] [**Software**: Auto-Throttle] in order to [**Goal**: Not Cause Auto-Throttle Disengagement].

B: If [**System**] [**Operator**] [**Value**] is [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] is [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____is____then____in order to____.

CR Type: Monitor

25. The FCCs shall issue a warning when the commanded altitude disagrees with the stored commanded altitude stored in the FCCs.

A: If [**Software**: FCC:Stored Commanded Altitude] [**Operator**: Not equal] [**Value**: Set Altitude] is [**Boolean**: True] then [**Software**: FCC] [**Soft-Human**: Will Warn] [**Human**: Pilot] in order to [**Goal**: Alert Pilot of Mismatch between Stored and Commanded Altitude].

B: If [**System**] [**Operator**] [**Value**] is [**Boolean**] then [**System**] [**Relationship**] [**System**] in order to [**Goal**].

C: If [**Test**] is [**Boolean**] then [**Response Action**] in order to [**Goal**].

D: If____is____then____in order to____.

CR Type: Monitor

26. The Auto-Throttle will be limited by the maximum throttle and the minimum throttle.

A: [**Software**: Auto-Throttle] is limited to [**Value**: [Min Throttle, Max Throttle]] in order to [**Goal**: Set Auto-Throttle Limits Correctly].

B: [**System**] is limited to [**Value**] in order to [**Goal**].

C: [**System**] is limited to [**Value**] in order to [**Goal**].

D: ____is limited to____in order to____.

CR Type: Constraint